A NEW POLYNOMIAL CUTTING-PLANE ALGORITHM FOR MAXIMUM WEIGHT MATCHINGS IN GENERAL GRAPHS

by

Michel Minoux*

* Centre National d'études des télécommunications, FRANCE

Conférence présentée au Centre de recherche sur les transports le 15 juin 1983

Université de Montréal
Centre de recherche sur les transports - Publication #302
Septembre 1982

Dépôt légal - 3e trimestre 1983

Abstract :

A new polynomial algorithm for maximum weight matchings in general (i.e. non bipartite) graphs is described. This algorithm is based on the solution of the continuous relaxation of the (integer) maximum weight matching problem, which is shown to be reducible to a maximum cost flow problem on an associated bipartite graph. This maximum cost flow problem is solved in $\mathcal{O}(N^3)$ time by means of a *double-labelling procedure* on the initial graph. When fractional solutions occur in the course of solving the continuous matching problem, then it is shown how the search for integer solutions can be resumed on a new graph derived from the initial graph by *shrinking* some odd subsets of vertices. Applying this result in a recursive fashion leads to a new algorithm for matching problems which can be viewed as a cutting-plane type procedure in which, at any step, the structure of a continuous matching problem is maintained for the current augmented linear program (the initial linear program plus the cuts). The overall polynomiality of the algorithm is then proved by showing that the total number of cuts generated is bounded above by N^2, where N is the number of vertices of the given initial graph.

The question of the existence of a polynomial cutting-plane algorithm for matching problems in graphs was raised by S. HOLM in 1980 who (improperly) conjectured the polynomiality of GOMORY's fractional algorithm (himself provided some counterexamples to the conjecture). To the author's knowledge, the approach described here provides the first example of a polynomial cutting-plane procedure, even for such a restricted class of integer linear programs as weighted matching problems in graphs.

1. - INTRODUCTION AND PROBLEM STATEMENT

Let $G=[X,U]$ be a simple connected unoriented graph, where $N=|X|$ is the number of vertices, and $M=|U|$ the number of edges.

A matching K of G is a subset of edges, no two of which have a common endpoint. A vertex $i \in X$ which is an endpoint of an edge of K will be said to be saturated for K ; ortherwise it is said to be unsaturated (or : exposed).

Now, we associate with each edge $u \in U$ a real number w_u called the weight of edge u . The weight of any subset $U' \subset U$ is : $w(U') = \sum_{u \in U'} w_u$.

In the sequel of this paper, we shall concentrate on the maximum weight matching problem on G , that is, the determination of a matching K^* such that, for any matching K of G : $w(K) \leq w(K^*)$.

We note that it is always possible to assume the weights strictly positive, since, starting from any matching K , the matching K' obtained by deleting the edges of weight ≤ 0 is such that $w(K') \geq w(K)$.

When the given graph G is bipartite, i.e, when X can be partitionned into 2 subsets of vertices X_1 and X_2 , and all the edges are of the form $u=(i_1,i_2)$ with $i_1 \in X_1$ and $i_2 \in X_2$, then (cf. FORD and FULKERSON 1962) the problem, (also known under the name : assignment problem) can be solved as a maximum cost flow problem between s (source) and t (sink) in the oriented graph Γ deduced from G in the following way :

- the set vertices of Γ is $X \cup \{s\} \cup \{t\}$;
- to each edge (i,j) of G $(i \in X_1, j \in X_2)$ corresponds an arc (i,j) of infinite capacity, and cost $c_{ij} = w_{ij}$;
- $|X_1|$ arcs are added of the form (s,i) (where $i \in X_1$) with capacity 1 and zero cost ; $|X_2|$ arcs are added, of the form (j,t) (where $j \in X_2$) with capacity 1 and zero cost.

By using the variation due to EDMONDS and KARP (1972) of the BUSACKER-GOWEN algorithm (cf. BUSACKER and GOWEN 1961) the problem can be solved in $O(N^3)$ time (N longest path computations on a graph of order $N+2$ where the lengths on the arcs are all non positive).

In the particular case of maximum cardinality matching ($w_u = 1$, $\forall u \in U$) HOPCROFT and KARP (1973) have described an $O(N^{5/2})$ algorithm.

For maximum cardinality matchings on arbitrary graphs (non bipartite) the solution method, first proposed by EDMONDS (1965) and the variations of it later on developped by BALINSKY (1969), GABOW (1973), and others, are based on the theory of alternating chains and on the theorem of BERGE (1957) which characterizes a maximum cardinality matching.

The method consists, at each step, in determining an alternating chain (or several vertex - disjoint alternating chains) along which the current matching can be augmented. When no such alternating chain exists, the current matching has maximum cardinality, by BERGE's theorem.

The most efficient implementations of this algorithm have complexity $O(N^{5/2})$ as in the bipartite case (cf. EVEN and KARIV 1975, BARTNIK 1978).

For the maximum weight matching problem, EDMONDS (1965) has shown how, by using the primal - dual algorithm of linear programming, it can be reduced to the solution of a sequence of cardinality matching problems. In addition, the algorithm (often referred to as the "blossom algorithm") provides a constructive proof of the EDMONDS' theorem characterizing the matching polytope. (This theorem is recalled in section 4.5 below). An $O(N^4)$ implementation of the blossom algorithm will be found, for instance, in GONDRAN and MINOUX (1979) chapter 7.

GABOW (1973) and LAWLER (1976) have described an $O(N^3)$ version of this algorithm at the price of a much more involved implementation.

By using a variation of the EDMONDS and JOHNSON's algorithm for the Chinese Postman Problem (see EDMONDS and JOHNSON 1973) DERIGS (1979) has described an efficient labelling procedure for determining an alternating chain of maximum (reduced) weight. When applied iteratively, this procedure produces, as formerly suggested by GLOVER (1967) the whole family of maximum weight matchings of cardinality 1, 2, ... k, ... The resulting complexity appears to be equivalent to that obtained by GABOW and LAWLER ($O(N^3)$) but the implementation of DERIGS' algorithm is somewhat easier.

The use of cutting-plane methods of integer programming for solving matching problems in graphs has been proposed by a number of authors. HOLM (1979) has investigated the application of GOMORY's fractional algorithm (GOMORY 1965) to matching problems and reports numerous computational experiments showing that the number of cuts necessary to get integral solutions is usually very low compared with the problem size (number of vertices). Even though no proof of the polynomiality of the algorithm could be given, HOLM's experiments have had the merit to show that cutting-planes algorithms could compete, in practice, with the other existing methods. PADBERG and RAO (1979) have studied the cut-generation problem : given any fractional solution obtained at some step of a cutting-plane algorithm, determine, among all the constraints defining the matching polytope (EDMONDS' cuts), a violated constraint. They describe a polynomial algorithm for solving this problem, thus opening the way to using standard linear programming softwares for the solution of matching problems.

The above-mentionned works by HOLM and PADBERG - RAO obviously raise a question of interest, from the theoretical as well as from practical point-of-view : is it possible to devise a polynomial cutting-plane algorithm for matching problems in graphs ? It is intended to show here, that it is possible to answer this question positively by evidencing new properties of matching problems and deriving from them a polynomial algorithm based on the continuous solution of the matching problem and the cutting-plane philosophy, though avoiding the use of the simplex algorithm. Schematically, we shall proceed by successively addressing the three main following issues :

- Question 1 : is the continuous matching problem (i.e. where the integrality requirements on the variables have been relaxed) polynomially solvable ?

- Question 2 : is it possible to implement a cutting-plane algorithm in such a way that, at each step, the resulting linear program (i.e. the original continuous problem plus the cuts) can be solved in polynomial time ?

- Question 3 : is the total number of cutting-planes generated in that way bounded by a polynomial in the problem size ?

The first question is answered in section 2, where it is shown that the continuous matching problem on an arbitrary graph is equivalent to a matching problem on a bipartite graph, hence to a maximum cost flow problem (this equivalence had already been noticed for the dual problem - the so-called vertex-packing

problem - by NEMHAUSER and TROTTER 1975, PICARD and QUEYRANNE 1977 among others). By using a variation of the BUSACKER - GOWEN algorithm and interpreting the procedure on the initial graph G, it will be shown that the continuous matching problem can be solved in $O(N^3)$ time by a double-labelling procedure on G.

Question 2 is addressed in section 3 where some relations existing between continuous and integer solutions to matching problems are studied. It will be shown how, under certain conditions, adding a cut to the continuous matching problem may lead to a problem of the same nature (though on a different graph derived from the initial one by shrinking some odd subset of vertices) thus opening the way to solving the "augmented" problem in polynomial time.

In section 4 it is shown how, by recursive application of the results obtained in § 3, an algorithm of the cutting-plane type can be derived in which, at each step, the "augmented" linear program is polynomially solvable, being equivalent to a continuous matching problem on some shrunken graph (deduced from the initial graph by successively shrinking odd subsets of vertices).

The solution of the (integer) maximum weight matching problem thus reduces to a sequence of maximum cost flow problems, hence (by using the double-labelling procedure of section 2) to a sequence of longest path problems.

Moreover, the total number of cutting-planes generated by the algorithm will be shown to be bounded by $\frac{N^2}{2}$ (this is the answer to question 3) thus leading to on overall complexity $O(N^4)$.

The algorithm is illustrated by means of an example, described in full detail, in section 4.6 and appears to be conceptually simpler and easier to implement than others previously known algorithms.

The notations used throughout this paper concerning graph theory are the standard ones (see BERGE 1970, GONDRAN and MINOUX 1979). In particular for a graph $G = [X, U]$:

$\omega(i)$ denotes the set of edges having i as an endpoint (the elementary cocycle associated with i).

2. - A POLYNOMIAL ALGORITHM FOR SOLVING THE CONTINUOUS MAXIMUM WEIGHT MATCHING PROBLEM

We show in this section that the continuous maximum weight matching problem in an arbitrary graph (non bipartite) is equivalent to a maximum weight matching problem on some associated bipartite graph (assignment problem) hence to a maximum cost flow problem. It is then shown in § 2.4 how the algorithm for solving the maximum cost flow problem can be actually implemented on the initial graph by means of a double-labelling procedure of complexity $O(N^2)$ which will be basic to the algorithm of section 4.

Several important properties relating the primal and dual solutions of the flow problem to the primal and dual solutions of the matching problems are stated here without proofs, since these are either straightforward or can be found elsewhere (see for instance NEMHAUSER and TROTTER 1975, PICARD and QUEYRANNE 1977).

2.1. - The matching problem and the associated continuous problem

Let $A = (a_{iu})_{\substack{i=1,\dots N \\ u=1,\dots M}}$ be the vertex-edge incidence matrix of graph G defined by : $a_{iu} = +1$ if i is an endpoint of u, $a_{iu} = 0$ otherwise. With each edge $u \in U$, we associate a 0 - 1 variable x_u. The vector $x = (x_u)_{u=1,\dots M}$ may thus be considered as the characteristic vector of the subset of edges $U_x = \{ u / u \in U ; x_u = 1 \}$.

It is well-known that U_x is a matching of G if and only if the system of inequalities

$$A.x \leq \mathbb{1}$$

is satisfied (here $\mathbb{1}$ denotes the N-vector with all components equal to 1). The maximum weight matching problem is then equivalent to the integer linear program :

$$(MP) \quad \begin{cases} \text{Maximize} \quad w.x = \sum_{u \in U} w_u . x_u \\ \text{Subject to} \\ \quad A.x \leq \mathbb{1} \\ \quad x_u = 0 \text{ or } 1 \quad (u = 1, 2, \dots M) \end{cases}$$

The continuous matching problem associated with (MP) is the linear program obtained by allowing any real value between 0 and 1 for each variable x_u, i.e. :

$$(CMP)\quad \begin{cases} \text{Maximize} \quad w.x \\ \text{Subject to :} \\ A.x \leq \mathbb{1} \\ 0 \leq x_u \leq 1 \quad (\forall u = 1,2,\dots M) \end{cases}$$

In the case of bipartite graphs (CMP) is known to be equivalent to (MP) since then the vertex-edge incidence matrix A is totally unimodular. In the case of non bipartite graphs, however, the optimal solution obtained for (CMP) will not be integer-valued, and the integrality requirements on the variables are compulsory. The algorithm for (MP) which will be described here relies on the solution of the continuous relaxation (CMP) which, as we now proceed to show, is equivalent to a maximum cost flow problem.

2.2. - Equivalence of the continuous matching problem with an assignment problem (matching on a bipartite graph)

To begin with, let us show that (CMP) is equivalent to a maximum weight matching problem in the bipartite graph $\bar{G} = [\bar{X}, \bar{U}]$ derived from G as follows :

- to each vertex i of G correspond two vertices of $\bar{G}$ which will be denoted by i' and i". If X' is the set of vertices of the form i' and X" the set of vertices of the form X", then $\bar{X} = X' \cup X''$.
- with each edge $u = (i,j)$ of weight w_u in G , we associate two edges $u' = (i', j'')$ and $u'' = (j', i'')$ of weight $\bar{w}_{u'} = \bar{w}_{u''} = w_u$ in $\bar{G}$. If U' (resp : U'') is the set of edges of the form u' (resp : of the form u''), then $\bar{U} = U' \cup U''$.

Let $\bar{A}$ be the vertex-edge incidence matrix of $\bar{G}$. By splitting the columns corresponding to the edges in U' from those corresponding to the edges in U'', we can write : $\bar{A} = [\bar{A}', \bar{A}'']$. Similarly, the weights $\bar{w}$ can be split into $\bar{w} = [\bar{w}', \bar{w}'']$. To each edge u' (resp : u''), associate a variable $\bar{x}_{u'}$ (resp : $\bar{x}_{u''}$) and denote by $\bar{x} = [\bar{x}_{u'}, \bar{x}_{u''}]$ the vector of variables thus obtained. The maximum weight matching problem on $\bar{G}$ is equivalent to the linear program :

$$(LP)\quad \begin{cases} \text{Maximize} \quad \bar{w}'.\bar{x}' + \bar{w}''.\bar{x}'' \\ \text{Subject to :} \\ \bar{A}'.\bar{x}' + \bar{A}''.\bar{x}'' \leq \mathbb{1} \\ \bar{x}' \geq 0 \quad ; \quad \bar{x}'' \geq 0 \end{cases}$$

In the above problem, the integrality requirements on the variables may be dropped since it is a bipartite matching problem.

The following property then shows how an optimal solution of (CMP) can be deduced from an optimal solution of (LP).

Property 1 :

Let $[\bar{x}', \bar{x}''] \geq 0$ *be an optimal primal solution of (LP) and* $[\bar{\pi}', \bar{\pi}'']$ *a corresponding optimal dual solution. Then :* $x^* = (x^*_u)_{u=1,\dots M}$ *defined by :* $\forall u : x^*_u = \frac{\bar{x}_{u'} + \bar{x}_{u''}}{2}$ *and* $\pi^* = (\pi^*_i)_{i=1,\dots N}$ *defined by :* $\forall i \in X : \pi^*_i = \frac{\bar{\pi}'_i + \bar{\pi}''_i}{2}$ *are optimal primal and dual solutions of (CMP) respectively.*

Since an optimal solution of (LP) has components 0 or 1 (bipartite matching) it follows the well-known fact that :

Property 2 :

(CMP) has always an optimal solution with components $0, +1$ *or* $+\frac{1}{2}$.

Also notice that, when an optimal solution x^* of (CMP) has fractional components, then its support (supp $(x^*) = \{u / u \in U ; x^*_u > 0\}$) generally contains an odd cycle (it is always possible to exclude the degenerate cases where the support contains an even cycle).

2.3. - Equivalence of (CMP) with a maximum cost flow problem

It follows from the above that (CMP) can be solved as a maximum cost flow problem on the graph $\bar{\Gamma}$ deduced from $\bar{G}$:

- by adding two vertices s (source) and t (sink) ;

- by replacing each edge of $\bar{G}$ of the form (i', j'') or (j', i'') by an arc (i', j'') or (j', i'') with infinite capacity and cost equal to the weight of the corresponding edge in G ;
- by adding N arcs of the form (s, i'), $i' \in X'$ of zero cost and capacity 1 ; and N arcs of the form (j'', t), $j'' \in X''$, of zero cost and capacity 1 ;
- by adding a return arc (t, s) with infinite capacity and zero cost.

Denote by $\varphi_{i',j''}$ (resp : $\varphi_{j',i''}$) the flow passing through the arcs (i', j'') (resp : (j', i'')) in an optimal solution of the network flow problem on $\bar{\Gamma}$. Then, in view of Property 1, the corresponding optimal primal solution of (CMP) is obtained by :

$$x^*_{ij} = \frac{\varphi_{i',j''} + \varphi_{j',i''}}{2}$$

Now, the way by which the optimal dual variables of (CMP) can be derived from the optimal potentials of the flow problem on $\bar{\Gamma}$ is stated in the following :

Property 3 :

Let $\bar{\omega}_{i'}$ $(i' \in X')$ and $\bar{\omega}_{j''}$ $(j'' \in X'')$ be the optimal potentials associated with the maximum cost flow problem on $\bar{\Gamma}$ (it always holds $\bar{\omega}_s = \bar{\omega}_t = 0$). Then optimal dual variables π^* are obtained for (CMP) by :

$$\forall i \in X : \quad \pi^*_i = \frac{\bar{\omega}_{i''} - \bar{\omega}_{i'}}{2}$$

2.4. - An algorithm for the continuous matching problem

For solving the maximum cost flow problem on $\bar{\Gamma}$, the algorithm of BUSACKER and GOWEN (1961) will be used, which consists in building up maximum cost flows of values 1, 2, 3, ... successively. At each step k, an optimum flow φ^k of value k is obtained from an optimum flow φ^{k-1} of value $k-1$ by a flow augmentation along a longest path from s to t in the incremental graph $\bar{\Gamma}(\varphi^{k-1})$ associated with φ^{k-1} (For a definition of the incremental graph, see below ; cf. also GONDRAN and MINOUX (1979) chapter 5).

The lengths of the arcs being either positive or negative in the incremental graph, the determination of a longest path would require an $O(N^3)$ computational complexity (for instance applying algorithms by FORD 1956 or DANTZIG, BLATTNER and RAO 1966). However EDMONDS and KARP (1972) have shown

how, by using the reduced costs (instead of the costs themselves), the problem can be reduced to finding a longest path with lengths all non positive, which requires only $O(N^2)$ time to be solved by an algorithm of the MOORE - DIJKSTRA type (MOORE 1957, DIJKSTRA 1959). Hence the solution of (CMP) can be found in $O(N^3)$ time complexity.

From an algorithmic point-of-view however, it is much preferable to directly operate on the initial graph G, instead of working on the auxiliary graph $\overline{\Gamma}$. To that purpose, it will be shown how to transpose the EDMONDS - KARP's algorithm and convert the search for a longest path in the incremental graph into a double-labelling procedure on G.

At the current step in the solution of (CMP) let $x = (x_u)_{u \in U}$ be the current primal solution (with components 0, 1 or $\frac{1}{2}$) and $\pi = (\pi_i)_{i \in X}$ the corresponding dual solution.

If $\sum_{u \in U} x_u = p$, x is of maximum weight among all the (continuous) solutions of (CMP) satisfying the additional constraint: $\sum_{u \in U} x_u = p$ (here p is an integer multiple of $\frac{1}{2}$).

In addition, π is a dual solution hence satisfies :

$$\forall u = (i,j) \in U : \quad \bar{w}_u = w_u - \pi_i - \pi_j \leq 0$$

($\bar{w}_u$ is called the reduced weight of edge u with respect to the dual solution π).

Moreover, we assume that the complementarity conditions (of type 1)

$$x_u > 0 \Rightarrow \bar{w}_u = 0$$

are satisfied (these will be maintained throughout the algorithm).

It follows that, if x and π are not optimal solutions of (CMP) and its dual, respectively, this means that one (or several) complementary condition(s) of type 2 : $\pi_i > 0 \Rightarrow$ vertex i is saturated for x is violated, hence that there exist unsaturated vertices $i \in X$ such that $\pi_i > 0$. For such unsaturated vertices, it is not restrictive to assume that all the associated dual variables are equal (if this is not so, replace π_i by $\tilde{\pi}_i = \max_{j \text{ unsaturated}} \{\pi_j\}$ for each unsatured vertex i). This condition will be maintained throughout the algorithm.

The solution x corresponds to a maximum cost flow of value $2p$ from s to t in the graph $\overline{\Gamma}$. The optimum flow of value $2p+1$ is then determined by looking for a longest path between s and t on the incremental graph associated with x, say $\overline{\Gamma}(x)$, and defined as follows :

- $\overline{\Gamma}(x)$ has the same set of vertices as $\overline{\Gamma}$ (the two vertices s and t and the $2N$ vertices i' and i'' corresponding to the vertices i of G) ;
- $\overline{\Gamma}(x)$ contains the arcs (i',j'') and (j',i'') of length $\ell(i',j'') = \ell(j',i'') = w_u$ whenever $u=(i,j)$ is an edge of G such that $x_u + \frac{1}{2} \leq 1$;
- $\overline{\Gamma}(x)$ contains the arcs (i'',j') and (j'',i') of length $\ell(i'',j') = \ell(j'',i') = -w_u$ whenever $u=(i,j)$ is an edge of G such that $x_u - \frac{1}{2} \geq 0$;
- $\overline{\Gamma}(x)$ contains arcs (s,i') and (i'',t) of zero length whenever i is unsaturated, and arcs (i',s) and (t,i'') of zero length whenever i is saturated.

It is seen that any path from s to t in $\overline{\Gamma}(x)$ has odd cardinality and successively encounters arcs of odd order and length $+w_u$, and arcs of even order and length $-w_u$ (where u is the corresponding edge of G).

Now, if we define the lengths of the arcs in $\overline{\Gamma}(x)$ in terms of the reduced weights $\overline{w}_u$ instead of the initial weights w_u, we get a longest path problem with all lengths (referred to in the sequel as "reduced lengths") non positive.

Indeed, for an odd order arc corresponding to the edge $u=(i,j)$ of G, the reduced length is $\overline{\ell}(i',j'') = \overline{\ell}(j',i'') = \overline{w}_u \leq 0$; and for an even order arc, the reduced length is $\overline{\ell}(i'',j') = \overline{\ell}(j'',i') = -\overline{w}_u = 0$ ($x_u \geq \frac{1}{2} > 0$ implies $\overline{w}_u = 0$ by virtue of the complementarity conditions).

Moreover, for any path P from s to t in $\overline{\Gamma}(x)$ the length $\ell(P)$ (w.r.t. the w_u) and the reduced length $\overline{\ell}(P)$ (w.r.t. the $\overline{w}_u$) are linked by the relation :

$$\ell(P) = \overline{\ell}(P) + 2\pi_{ns}$$

(where π_{ns} is the dual variable associated with the non saturated vertices). Thus P is of maximum length if and only if is of maximum reduced length.

The reduced lengths all being non positive, an algorithm of the MOORE - DIJKSTRA type ($O(N^2)$) may then be used to solve the longest path problem on $\overline{\Gamma}(x)$. However, the operations will be on the initial graph G itself by associating

with each vertex $i \in X$ a double label $[m'_i, m''_i]$ which can be thought of as the labels of the corresponding vertices i' and i'' of $\overline{\Gamma}$. m'_i will be called the even label and m''_i the odd label of vertex i.

The MOORE - DIJKSTRA algorithm on $\overline{\Gamma}(x)$ then specializes to the following double-labelling procedure on G :

Procedure 1 (double labelling)

(a) *initializing the labels : set :*
$m'_i = 0$, *if* i *is unsaturated ;* $m'_i = -\infty$ *otherwise ;* $m''_i = -2\pi_{ns}$ *if* i *is saturated ;* $m''_i = -\infty$ *otherwise (* π_{ns} *is the common value of the dual variables associated with unsaturated vertices).*
Set $I' = I'' = X$

(b) *If* $|I'| = |I''| = 0$: *TERMINATE. (All vertices have received a permanent even label and a permanent odd label). Go to (c).*
Otherwise :

Compute $m_{max} = \max \left\{ \max_{i \in I'} \{m'_i\} ; \max_{i \in I''} \{m''_i\} \right\}$

Two situations may occur :

Case 1 : $m_{max} = m'_i$ *for some* $i \in I'$.
Then m'_i *becomes permanent even label for* i . *For all the edges* $u = (i,j) \in \omega(i)$ *such that* $x_u \leq \frac{1}{2}$ *and* $j \in I''$:
- *compute* $\alpha = m'_i + \bar{w}_u$
- *if* $m''_j < \alpha$ *set* $m''_j \leftarrow \alpha$

Put $I' \leftarrow I' - \{i\}$ *and return to (b)*

Case 2 : $m_{max} = m''_i$ *for some* $i \in I''$
Then m''_i *becomes permanent odd label for* i . *For all the edges* $u = (i,j) \in \omega(i)$ *such that* $x_u \geq \frac{1}{2}$ *and* $j \in I'$:
- *compute* $\alpha = m''_i - \bar{w}_u$ $(= m''_i$ *since* $\bar{w}_u = 0)$
- *if* $m'_j < \alpha$ *set* $m'_j \leftarrow \alpha$

Put $I'' \leftarrow I'' - \{i\}$ *and return to (b)*

(c) *At the end of the procedure, the labels* m''_i *represent the reduced lengths of the longest paths between* s *and* i'' *on* $\overline{\Gamma}(X)$. *Thus the required optimal path has length :*

$$\max_{i \text{ unsaturated}} \{ m''_i + 2\pi_{ns} \}$$

Once the longest path has been determined, a flow augmentation on $\overline{\Gamma}(x)$ has to be performed, which corresponds on G to a change of the primal solution x. The longest path on $\overline{\Gamma}(x)$ induces on G an alternating chain (not neccessarily elementary) alternatively passing through odd edges (such that $x_u + \frac{1}{2} \leq 1$) and even edges (such that $x_u - \frac{1}{2} \geq 0$) and joining two unsaturated vertices i and j (where possibly $i = j$). Then, one of the two following situations may occur :

Case 1 :

L is an elementary alternating chain along which either $x_u = 0$ or $x_u = 1$, and joining two distinct unsaturated vertices i and j. In this case, the new primal solution x' is obtained by a transfer along L, that is, by interchanging the 0's and the 1's along the chain. In this transformation, the cardinality of the solution is increased by 1.

Case 2 :

L is either a non elementary alternating chain (hence containing an odd cycle) ; or contains edges u such that $x_u = \frac{1}{2}$. In this case, an odd cycle may be either created or destroyed, but the cardinality of the solution (the quantity $\sum_{u \in U} x_u$) is only increased by $\frac{1}{2}$. The new primal solution x' is obtained by increasing (resp : decreasing) the value x_u by $\frac{1}{2}$ on the odd edges (resp : on the even edges) of the chain L.

Once the new primal solution has been determined, the new dual solution can be deduced from the labels $[m'_i , m''_i]$ obtained at the end of Procedure 1, as the following property shows :

Property 4 :

At the end of Procedure 1 a new dual solution π' is given by :

$$\begin{cases} \pi'_i = \pi_i + \dfrac{m''_i - m'_i}{2} & \text{for a saturated vertex } i \\ \pi'_i = \pi_i + \max\limits_{j \text{ unsaturated}} \left\{ \dfrac{m''_j}{2} \right\} & \text{for an unsaturated vertex } i \end{cases}$$

and satisfies the complementarity conditions (of type 1) with the new primal solution x'.

We also notice that the above transformation maintains equality between the dual variables associated with the unsaturated vertices.

To summarize, the solution algorithm for the continuous matching problem (CMP) on G can be stated as follows :

Algorithm 1 (solving (CMP) on G)

(a) Initialization :

$x = 0$

$\pi_i = \frac{1}{2} \max_{u \in U} \{ w_u \} \quad (\forall i \in X)$

Iteration $\ell = 0$.

(b) $\ell \leftarrow \ell + 1$

At iteration ℓ , *let* x *be the current primal solution,* π *the corresponding dual solution. Determine a longest path for augmentation by double-labelling on* G *(Procedure 1). For* $i \in X$, *let* $[m'_i , m''_i]$ *be the labels obtained.*

(c) Determine the new dual solution π' *by :*

$$\pi'_i = \pi_i + \frac{m''_i - m'_i}{2} \qquad (i \text{ saturated})$$

$$\pi'_i = \pi_i + \max_{j \text{ unsaturated}} \left\{ \frac{m''_j}{2} \right\} \qquad (i \text{ unsaturated})$$

(d) Let $\delta = \max_{i \text{ unsaturated}} \{ m''_i + 2\pi_i \}$

the length of the longest path obtained.

If $\delta \leq 0$, *TERMINATE :* (x, π') *is a pair of optimal solutions to the primal and to the dual respectively.*

If $\delta > 0$, *determine a new primal solution* x' *(case 1, case 2)*
Set $(x, \pi) \leftarrow (x', \pi')$ *and return to (b).*

We notice the following property which will be used subsequently :

Property 5 :

In the course of Algorithm 1 the dual variables associated with the unsaturated vertices are monotonically non increasing.

proof :

The m''_j represent lengths of optimal paths in a graph with non positive lengths on the arcs, hence for an unsaturated vertex i , it always holds $\pi'_i \leq \pi_i$. □

Remark

Algorithm 1 solves (CMP) by successively solving cardinality constrained continuous matching problems of the form :

$$(CMP[p]) \quad \begin{cases} \text{Maximize} \quad w.x \\ A.x \leq \mathbb{1} \\ \mathbb{1}.x = p \\ 0 \leq x_u \leq 1 \quad (\forall u \in U) \end{cases}$$

for $p = 0, \frac{1}{2}, 1, \frac{3}{2}, \ldots$

We then observe that it is easy to deduce, from the dual solution obtained by Algorithm 1 (when the cardinality is p) a dual optimal solution of (CMP $[p]$).

Denote by $\underset{\sim}{\pi} = (\underset{\sim}{\pi}_i)_{i \in X}$ the dual variables associated with the constraints $A.x \leq \mathbb{1}$ in (CMP $[p]$) and by λ the dual variable associated with the cardinality constraint $\sum_{u \in U} x_u = p$, then a dual optimal solution of (CMP $[p]$) is :

$$\begin{cases} \forall i \in X : & \underset{\sim}{\pi}_i = \pi_i - \pi_{ns} \\ & \lambda = 2\pi_{ns} \end{cases}$$

where π_{ns} is the dual variable of an unsaturated vertex.

This remark will be used in the proof of theorem 1 (section 3.3.). ||

3. - RELATIONS BETWEEN CONTINUOUS AND INTEGER OPTIMAL SOLUTIONS OF MATCHING PROBLEMS

This section is devoted to the study of some relations linking continuous and integer optimal solutions of matching problems in graphs.

The main result stated and proved in § 3.3. shows that, if the optimal continuous solution on G for some (integer) cardinality p is not integer, then an optimal integer solution of cardinality p can be deduced from an optimal integer solution on a graph G_1 derived from G by shrinking some odd subset of vertices. This is, to our knowledge, a new result generalizing to weighted matching problems, a classical result used for solving cardinality matching problems (see eg. LAWLER 1976, or GONDRAN and MINOUX 1979 chap. 7 Lemma 3). This will be used in section 4 to devise a new algorithm for maximum weight matching problems.

3.1. - Necessary conditions for a continuous solution to be a matching : EDMONDS' constraints

For any subset of vertices $S \subset X$, we denote by U_S the subset of edges with both endpoints in S.

Obviously, any matching $K \subset U$ must satisfy the relation :

$$|K \cap U_S| \leq f_S = \frac{|S|-1}{2}$$

for any odd subset of vertices S. This is because the subset of edges $K_S = K \cap U_S$ is a matching in the subgraph $G_S = [S, U_S]$, and its cardinality cannot be greater than $\frac{|S|-1}{2}$.

Let x be an integer solution of the system (1) $A.x \leq 1$ (i.e. the characteristic vector of a matching in G). Denoting by F_S the 0 - 1 characteristic vector of U_S in U, it follows from the above that x satisfies all the constraints (often referred to as the EDMONDS' constraints) :

$$F_S \cdot x \leq f_S \qquad \ldots (2)$$

(for all odd subsets of vertices S).

The basic principle, underlying all existing solution methods for weighted matching problems consists in appending to (CMP) all the EDMONDS' constraints and to solve the augmented linear program (in continuous variables) :

$$(CMP)' \begin{cases} \text{Maximize} \quad w.x \\ \text{Subject to} \\ A.x \leq \mathbb{1} \qquad \ldots (1) \\ F_S.x \leq f_S \qquad (\forall S \subset X, |S| \text{ odd}) \qquad \ldots (2) \\ x \geq 0 \qquad \ldots (3) \end{cases}$$

and finally to show that an optimal integer solution (an optimum matching) is necessarily obtained.

This result, first proved by EDMONDS (1965), shows that the extreme points of the polytope defined by (1) - (2) - (3) (the so-called matching polytope) correspond one-to-one to matchings in G. It is then both necessary and sufficient to add constraints (2) to (CMP) to eliminate non integer optimal solutions.

For solving (CMP)' EDMONDS (1965) described an algorithm (derived from the primal - dual algorithm of linear programming) in which the problem is reduced to the solution of a sequence of maximum cardinality matching problems. We suggest here a new approach consisting in using the solution of the continuous matching problem (Algorithm 1 of § 2.4.) according to a cutting-plane strategy.

3.2. - Necessary and sufficient optimality conditions for (CMP)'

In the following, we assume that the odd subsets of vertices are numbered (in an arbitrary fashion) and indexed by k, k running from 1 to k_{max} (the number of odd subsets of X).

Associating with each vertex $i \in X$ (with each row of A) a dual variable $\pi_i \geq 0$ and with each constraint (2) of the form :

$$F_k.x \leq f_k$$

a dual variable $\mu_k \geq 0$, the dual of (CMP)' can be written as :

$$(D') \begin{cases} \text{Minimize} & \sum_{i=1}^{N} \pi_i + \sum_{k=1}^{k_{max}} \mu_k \cdot f_k \\ \text{Subject to} & \\ & \sum_{i=1}^{N} \pi_i A_i + \sum_{k=1}^{k_{max}} \mu_k \cdot F_k \geq w \qquad \ldots (4) \\ & \pi \geq 0 \ , \ \mu \geq 0 \end{cases}$$

(A_i is the i^{th} row of A) or in obvious matrix notations :

$$(D') \begin{cases} \text{Minimize} & \pi . \mathbb{1} + \mu . f \\ \text{Subject to :} & \\ & \pi . A + \mu . F \geq w \\ & \pi \geq 0, \ \mu \geq 0 \end{cases}$$

It is a well-known result in linear programming that x (solution of (CMP)') and (π, μ) (solution of (D')) are both optimal solutions in the primal and in the dual respectively, if and only if they satisfy the so-called <u>complementary conditions</u> :

$$(C) \begin{cases} \text{(C1) for each edge } u = (i,j) \in U : & \\ \qquad x_u > 0 \Rightarrow \bar{w}_u = w_u - (\pi_i + \pi_j + \sum_{k \in \chi(u)} \mu_k) = 0 & \\ \text{(C2)} \quad \pi . (A . x - \mathbb{1}) = 0 & \\ \text{(C3)} \quad \mu . (F . x - f) = 0 \end{cases}$$

Where $\chi(u)$ denotes the set of indices k of odd subsets of vertices containing both endpoints of u .

By definition, $\bar{w}_u$ above will be called the <u>reduced weight</u> of edge u (with respect to the dual variables (π, μ)).

Thus the solution of (CMP)' amounts to determining :

(a) dual variables (π, μ) such that, for any edge $u = (i,j) \in U$:

$$\bar{w}_u = w_u - \pi_i - \pi_j - \sum_{k \in \chi(u)} \mu_k \leq 0$$

(the reduced weights are all non positive).

(b) primal variables x satisfying (1) (2) (3) and meeting the complementary conditions $(\mathcal{C})$ with (π, μ) .

3.3. - Relation between continuous and integer optimal solutions

Suppose we solve on the initial graph $G_0 = G$ the problems

$$(CMP[p]) \begin{cases} \text{Maximize } w.x \\ A.x \leq \mathbb{1} \\ \mathbb{1}.x = p \\ 0 \leq x_u \leq 1 \quad (\forall u \in U) \end{cases}$$

for $p = 1, 2, \ldots$ sucessively.

Algorithm 1 of § 2.4. can be used to solve (CMP [1]), and then, starting from the solution obtained, (CMP [2]) and so on. As long as the solutions obtained have integer 0 - 1 components, these are obviously optimal solutions of the corresponding cardinality - constrained (integer) matching problems :

$$(MP[p]) \begin{cases} \text{Maximize } w.x \\ A.x \leq \mathbb{1} \\ \mathbb{1}.x = p \\ x_u = 0 \text{ ou } 1 \quad (\forall u \in U) \end{cases}$$

Now, suppose that this holds true for $p = 1, 2, \ldots q$ (q integer) but that the solution of (CMP $[q+\frac{1}{2}]$) contains an odd cycle γ_1 in its support. Let $S_1 \subset X$ be the set of vertices of γ_1 , and $G_1 = G_0 / S_1$ be the graph obtained by shrinking the subset of vertices S_1 (shrunken graph). By definition, G_1 is derived from G_0 : (i) by replacing all the vertices in S_1 by a single vertex s_1 , say ; (ii) by replacing each edge $u = (i, j)$ of G_0 such that $i \notin S_1$, $j \in S_1$ by an edge $u = (i, s_1)$ in G_1 (the edges in U_{S_1} do not belong any longer to G_1).

A vertex such as s_1 in G_1 , obtained by shrinking an odd subset of vertices of G_0 , will be called a pseudo-vertex of G_1 . As opposed to pseudo-vertices, the vertices in G_1 corresponding to vertices of the initial graph G_0 will be called true vertices.

Let $\bar{\pi}^0_j$ $(j \in X)$ be the dual variables obtained by Algorithm 1 when solving (CMP $[q+\frac{1}{2}]$) and $\bar{\pi}^0_{min} = \min_{j \in S_1} \{\bar{\pi}^0_j\}$.

With the edges of $G_1 = G_0/S_1$, we now associate the weights $w^1 = w^0/S_1$ defined in the following way :

$$\begin{cases} w^1_u = w^0_u \ (= w_u) & \text{for any edge } u \text{ which is not incident to } s_1 \text{ in } G_1 ; \\ w^1_u = w^0_u - (\bar{\pi}^0_j - \bar{\pi}^0_{min}) & \text{for an edge } u = (i, s_1) \text{ incident to } s_1 \text{ in } G_1 \text{ and deriving from an edge } u = (i,j) \text{ in } G_0 . \end{cases}$$

Let $f_1 = \frac{|S_1| - 1}{2}$. Theorem 1 below then shows that the search for an optimum (integer) matching of cardinality $q+1$ on G_0 may be resumed by looking for an optimum matching of cardinality $q - f_1 + 1$ on G_1 , where the edges have weights $w^1 = w^0/S_1$.

Theorem 1 :

Suppose that $\bar{x}^1$ is an optimal solution of (CMP $[q - f_1 + 1]$) on G_1 with all integer components (it is thus a maximum weight matching of cardinality $q - f_1 + 1$ on G_1) and let $\bar{\pi}^1$ be the dual solution given by Algorithm 1 and correspondind to $\bar{x}^1$.

Then, if $\bar{\pi}^1_{s_1} \leq \bar{\pi}^0_{min} = \min_{i \in S_1} \{\bar{\pi}^0_i\}$, $\bar{x}^1$ induces on G_0 an optimal integer solution (an optimal matching) of cardinality $q+1$.

Proof :

Let $(\bar{\pi}^1, \lambda)$ be the optimal dual solution of (CMP $[q - f_1 + 1]$) on G_1 , where λ is the dual variable associated with the cardinality constraint. From § 2.4. $(\bar{\pi}^1, \lambda)$ is deduced from $\bar{\pi}^1$ by :

$$\begin{cases} \lambda = 2\, \bar{\pi}^1_i & \text{(where } i \text{ is an unsaturated vertex)} \\ \underset{\sim}{\bar{\pi}}{}^1_i = \bar{\pi}^1_i - \frac{\lambda}{2} & (\forall i \in X) \end{cases}$$

For vertex s_1 we notice that :

$$\underset{\sim}{\bar{\pi}}{}^1_{s_1} + \frac{\lambda}{2} \leq \bar{\pi}^0_{min}$$

(this is so because $\bar{\pi}^1_{s_1} + \frac{\lambda}{2} = \bar{\pi}^1_{s_1}$ and $\bar{\pi}^1_{s_1} \leq \bar{\pi}^0_{min}$ by the assumption of the theorem).

Now, it follows from the general theory of duality in convex programming (see MINOUX 1979 for instance) that $\bar{x}^1$ and $\bar{\pi}^1$ are respectively primal and dual optimal solutions to the continuous matching problem (whithout cardinality constraint) :

$$(\widehat{CMP}) \begin{cases} \text{Max } \hat{w}.x = (w^1 - \lambda \mathbb{1}).x \\ \text{Subject to :} \\ A^1.x \leq \mathbb{1} \\ 0 \leq x_u \leq 1 \quad (\forall u) \end{cases}$$

(A^1 denotes the vertex-edge incidence matrix of G_1). In other words, $\bar{x}^1$ is an optimal solution of the continuous matching problem on G_1, the edges u of G_1 being assigned weights $w^1_u - \lambda$.

Let x^0 be the solution induced by $\bar{x}^1$ on G_0. x^0 is defined as follows :

- If the vertex s_1 is <u>unsaturated</u> with respect to $\bar{x}^1$ in G_1, then the induced matching on G_0 is obtained by adding to $\bar{x}^1$ a maximum cardinality matching of the odd cycle γ_1 leaving exposed a vertex j_0 of S_1 such that $\bar{\pi}^0_{j_0} = \bar{\pi}^0_{min}$.

- If the vertex s_1 is <u>saturated</u> with respect to $\bar{x}^1$ (i.e. $\bar{x}^1_u = 1$ for some edge u incident to s_1) then we add to $\bar{x}^1$ the maximum cardinality matching of γ_1 leaving exposed the endpoint of u in S_1 .

Now consider the following dual solution (π^0, μ^0) defined by :

$$\pi^0_i = \bar{\pi}^1_i \quad \text{for } i \in X - S_1$$
$$\pi^0_j = \bar{\pi}^0_j - (\bar{\pi}^0_{min} - \bar{\pi}^1_{s_1}) \quad \text{for } j \in S_1$$
$$\mu^0_1 = 2\left(\bar{\pi}^0_{min} - \bar{\pi}^1_{s_1} - \frac{\lambda}{2}\right)$$

It can be checked (the details of this are left to the reader) that (π^0, μ^0) together with x^0 satisfy all the complementary conditions (C1) (C2) (C3) of § 3.2. for the matching problem on G_0 where the edges u are assigned weights $w^0_u - \lambda$. (In particular, observe that $\mu^0_1 \geq 0$ since $\bar{\pi}^1_{s_1} + \frac{\lambda}{2} \geq \bar{\pi}^0_{min}$). It follows that x^0 is an optimum matching on G_0 with cardinality $q+1$ (and λ is the dual variable associated with the cardinality constraint). □

Notice that, obviously, Theorem 1 could be stated in slightly more general terms : optimal (integer) solutions of cardinality $q - f_1 + 2$, $q - f_1 + 3$, ... etc. on G_1 (and such that the dual variable associated with δ_1 remains less or equal to $\bar{\pi}^{\circ}_{min}$) induce on G_o optimal integer solutions of cardinality $q+2$, $q+3$, ... etc.

Theorem 1 is an extension of an earlier result stated in MINOUX (1980) for the case where cardinality constraints are not considered (for a complete proof see MINOUX 1982).

4. - A NEW ALGORITHM FOR MAXIMUM WEIGHT MATCHING PROBLEMS IN GENERAL GRAPHS

In this section, we describe a new algorithm for maximum weight matching which is based on a recursive application of theorem 1 of § 3.3. This algorithm, which uses at each step the solution of some continuous matching problems, may be viewed as a cutting-plane method. However, contrary to the other cutting-plane algorithms suggested in the litterature (cf. HOLM 1979, PADBERG and RAO 1979) its efficiency essentially stems from the fact that it avoids the simplex method of linear programming (which, among other drawbacks, should be very sensitive here to degeneracies) by exploiting the reduction of the continuous matching problem to a maximum cost flow problem.

After an exposition of the basic ideas in § 4.1., a detailed statement of the algorithm is given in § 4.2. It proceeds in, at most, $\frac{N}{2}$ steps, each step p ending by producing a maximum weight matching of cardinality p . Thus, just as the algorithm by WHITE (1967) or the one by DERIGS (1979), it generates the complete family of optimum matchings for all possible cardinalities. Finite convergence is proved in § 4.3. and an $O(N^4)$ complexity bound is obtained by showing that each step requires at most N longest paths computations in a graph with $2N$ vertices and lengths all non positive.

A detailed illustration of how the algorithm works on an example will be found in section 4.6.

It is interesting to note, here, that the complexity bound given above can be significantly improved by exploiting some more properties of the algorithm and by using appropriate data structures. An $O(N^3)$ version of the algorithm can thus be obtained, and, when specializing to sparse graphs (i.e. graphs such that $M << \frac{1}{2}N(N-1)$) the complexity can be further improved to $O(MN\sqrt{N})$. For the details of the discussion concerning complexity improvements, we refer to MINOUX (1982).

However, though the $O(N^4)$ version of the algorithm described in this paper does not achieve the best possible complexity bound, it will be seen to be somewhat easier to implement than other existing methods, since it uses basically classical network flow and shortest (longest) paths subroutines.

4.1. - Outline of the algorithm

The algorithm basically consists in a recursive application of Theorem 1 of § 3.3. It works as follows.

Algorithm 1 of § 2.4. is applied on the initial graph $G_0 = G$ as long as the solutions obtained remain integer, i.e. as long as the changes in the primal solution occur along odd elementary alternating chains (Case 1). Thus optimal solutions of problems (MP[p]) are obtained for successive values $p = 1, 2, \cdots q$.

As soon as case 2 (cf. § 2 4.) is encountered, Algorithm 1 is interrupted at the end of step (c), after the change in dual solution has been performed, but without changing the primal solution. Indeed this corresponds to the situation where the optimal solution of (CMP$[q+\frac{1}{2}]$) would not be integer, i.e. would contain an odd cycle γ_1 in its support. (The odd cycle γ_1 is determined by tracing the longest path obtained by the double-labelling procedure).

Theorem 1 then shows that the search for optimal integer solutions of cardinality $q+1, q+2, \cdots$ can be resumed by applying Algorithm 1 on the shrunken graph $G_1 = G_0 / S_1$ (where S_1 is the vertex set of γ_1). More precisely, if $\bar{x}^0$ is the (integer) solution of (CMP$[q]$) obtained on G_0 and $\bar{\pi}^0$ the dual solution produced by Algorithm 1 for the cardinality $q+\frac{1}{2}$, the edges of G_1 will be assigned weights $w^1 = w^0 / S_1$ defined (as in § 3.3.) by :

$$\begin{cases} w^1_u = w^0_u & \text{for an edge } u \text{ which is not incident to } s_1 \text{ in } G_1 ; \\ w^1_u = w^0_u - (\bar{\pi}^0_j - \bar{\pi}^0_{min}) & \text{for an edge } u \text{ incident to } s_1 \text{ and deriving from an edge } u = (i,j) \text{ of } G_0 \ (i \notin S_1 ; j \in S_1). \end{cases}$$

Algorithm 1 will then be applied on G_1, taking as a starting point :

- the primal solution consisting in the restriction of $\bar{x}^0$ to G_1 ;
- the dual solution induced by $\bar{\pi}^0$ on G_1 , i.e. the solution π' defined by :

$$\begin{cases} \pi'_i = \bar{\pi}^0_i & \text{for } i \in X - S_1 \\ \pi'_{s_1} = \bar{\pi}^0_{min} = \min\limits_{j \in S_1} \{ \bar{\pi}^0_j \} & \text{for pseudo-vertex } s_1 \end{cases}$$

and the procedure is continued until one of the three following situations is met :

(A) The current (integer) solution on G_1 is optimal (the last longest path obtained has length $\zeta \leq 0$) ;

(B) Case 2 (cf. § 2.4.) occurs in the application of Algorithm 1 ;

(C) The dual variable associated with the pseudo-vertex s_1 becomes strictly greater than $\bar{\pi}^0_{min}$.

In situation (A), it follows from Theorem 1 that the optimal integer solution on G_1 induces an optimum matching on G_0 and the algorithm terminates.

In situation (B) a new odd cycle γ_2 has been detected. If S_2 denotes the set of its vertices, application of Algorithm 1 will have to be resumed on the shrunken graph $G_2 = G/S_2$. The weights $w^2 = w^1/S_2$ assigned to the edges of G_2 are defined in the same way as in § 3.3., using the dual variables given by Algorithm 1 after the cycle γ_2 has been found, and the change in dual variables has been performed.

Finally, we consider situation (C).

Suppose that the dual solution π' obtained on G_1 at the previous iteration does satisfy $\pi'_{s_1} \leq \bar{\pi}^0_{min}$, but that, at the current iteration, the new dual solution π'' which would result from the change in dual variables would satisfy $\pi''_{s_1} > \bar{\pi}^0_{min}$.

In this case, it is necessary to expand the pseudo-vertex s_1 , that is return back to the situation before the odd subset S_1 was shrunk.

Indeed, this means that the constraint $F_1 . x \leq f_1$ corresponding to S_1 (cf. § 3.3.) is no longer binding, in other words, that the associated dual variable μ_1 must become zero.

Since μ_1 is defined by : $\mu_1 = \ell(\bar{\pi}^0_{min} - \pi_{s_1})$ the dual variable π_{s_1} has to be equal to $\bar{\pi}^0_{min}$. Thus, before expanding the pseudo-vertex s_1 , a new dual solution π will be determined such that $\pi_{s_1} = \bar{\pi}^0_{min}$. Such a solution is given by :

$$\pi = (1-\theta)\,\pi' + \theta\,\pi''$$

where θ $(0 \leq \theta \leq 1)$ is a real parameter defined by :

$$\bar{\pi}^{\circ}_{min} = (1-\theta).\pi'_{\delta_1} + \theta.\pi''_{\delta_1}$$

Algorithm 1 will then be continued on the expanded graph obtained :

- by replacing pseudo-vertex δ_1 by all the vertices in S_1 ;
- by restoring the weights of the edges between vertices in S_1, or joining vertices of S_1 to vertices in $X - S_1$, to their respective values before shrinking S_1 ;

taking as starting solutions :

- the primal solution induced on the expanded graph by the current primal solution on G_1 (the induced primal solution is defined as in the proof of Theorem 1)
- the dual solution induced on the expanded graph by $\pi = (1-\theta)\pi' + \theta\pi''$, namely : the dual variables are set to π_i for the vertices not belonging to S_1 ; they are set to $\bar{\pi}^{\circ}_j$ for the vertices j in S_1 .

More generally, consider an arbitrary step of the process in which Algorithm 1 is applied to some current graph G_c derived from $G_0 = G$ by successive shrinking and expansions. The graph G_c contains a number of pseudo-vertices $\delta_1, \delta_2, \cdots$ resulting from the shrinking of some odd subsets of vertices (pseudo vertices or true vertices). To each pseudo-vertex δ of G_c, obtained by shrinking a subset of vertices S, we can associate a number $\pi_{min}(\delta)$ equal to the minimum of the dual variables associated with the vertices of S at the time when δ was created.

Moreover, since Algorithm 1 on G_c should be interrupted in each of the three situations (A) (B) (C) studied above, it is actually in the (slightly modified) form of Algorithm 1' below that it will be applied to G_c (only a typical step p is described).

Algorithm 1' (Step p)

(a) Let G_c be the current shrunken graph, x the current (integer) primal solution on G_c, π the current dual solution.

Beginning of step p . (p *is the cardinality of the solution induced by* x *on the initial graph* $G_0 = G$) .

(b) *Look for a longest path between* s *and* t *on the incremental graph* $\overline{\Gamma}_c(x)$ *by double labelling on* G_c *(Procedure 1).*

For any vertex i *of* G_c , *let* $[m'_i , m''_i]$ *be the labels obtained and :*

$$\gamma = \max_{i \text{ unsaturated}} \{ m''_i + \ell \pi_i \}$$

the length of the longest path obtained.

(c) *Let :*

$$\begin{cases} \Delta \pi_i = \dfrac{m''_i - m'_i}{\ell} & \text{for } i \text{ saturated} \\ \Delta \pi_i = \max\limits_{j \text{ unsaturated}} \left\{ \dfrac{m''_j}{\ell} \right\} & \text{for } i \text{ unsaturated} \end{cases}$$

If, for all pseudo-vertices Δ *of* G_c *it holds :*

$$\pi_\Delta + \Delta \pi_\Delta \leq \pi_{\min}(\Delta)$$

then set : $\pi \leftarrow \pi + \Delta \pi$ *and go to (d); otherwise, let* $\hat{\theta}$ *be the largest* θ *such that* $\pi_\Delta + \theta \, \Delta \pi_\Delta \leq \pi_{\min}(\Delta)$ *for all pseudo-vertices* Δ *of* G_c . *Hence* $\pi_{\Delta_0} + \hat{\theta} \, \Delta \pi_{\Delta_0} = \pi_{\min}(\Delta_0)$ *for some pseudo-vertex* Δ_0 .

Set $\pi \leftarrow \pi + \hat{\theta} \, \Delta \pi$ *and stop (EXIT "C" with indication of the pseudo-vertex* Δ_0 *to be expanded).*

(d) . *If* $\gamma \leq 0$ *TERMINATE (EXIT "A")*

. *If* $\gamma > 0$ *and if the longest path obtained in (b) contains an odd cycle* γ *(Case 2 of § 2.4.) then stop (EXIT "B" with indication of the odd cycle* γ *to shrink).*

. *If* $\gamma > 0$ *and if the longest path obtained in (b) corresponds to an elementary alternating chain, L, (Case 1 of § 2.4.), then obtain a new (integer) primal solution* x' *by performing a transfer along L, thus increasing cardinality by 1.*

End of Step p

It is seen that integrality of the solutions is maintained in Algorithm 1' since it is interrupted (EXIT "B") as soon as Case 2 of § 2.4. occurs.

We now describe a solution algorithm for the maximum weight matching problem which proceeds by repeated application of Algorithm 1'.

4.2. - The new algorithm

Algorithm 2 (maximum weight matching on arbitrary graphs)

(a) Set $G_c = G$ (the initial graph)

$x = 0$ (starting primal solution)

$\forall i \in X : \quad \pi_i = \frac{1}{2} \underset{u \in U}{Max} \{ w_u \}$ (starting dual solution)

Step $p = 0$

(b) At the current step p, let x be the current primal solution on G_c (it induces an optimum matching of cardinality p on the initial graph) and π the current dual solution.

(c) Apply Algorithm 1', starting from (x, π) on G_c, until one of the following situations is met :

(c1) Algorithm 1' terminates on EXIT "A". END : the current solution induces a maximum weight matching on the initial graph (the cardinality of which is p).

(c2) Algorithm 1' terminates on EXIT "B" and γ is the odd cycle to be shrunk.

Let $G'_c = G_c / S$ where S is the set of vertices (true vertices or pseudo-vertices) of γ . Let s be the pseudo-vertex of G'_c corresponding to S . Set $\pi_{min}(s) = \underset{i \in S}{Min} \{ \pi_i \}$.

Let $w' = w/S$ be the edge weights on G'_c (the weight is unchanged for all edges not incident to s ; for an edge u incident to s and deriving from edge (i,j) of G_c $(i \notin S, j \in S)$ the weight is decreased by $\pi_j - \pi_{min}(s)$). The new current primal solution x' is the restriction of x to the edges of G'_c . The new dual solution π' is defined by assigning the value $\pi'_s = \pi_{min}(s)$ to pseudo-vertex s , the dual variables of the other vertices of G'_c being unchanged.

Set : $G_c \leftarrow G'_c$; $w \leftarrow w'$; $x \leftarrow x'$;
$\pi \leftarrow \pi'$ *and return to (c).*

(c3) *Algorithm 1' terminates on EXIT "C" and* S_o *is the pseudo-vertex to be expanded.*

Let G'_c *be the expanded graph (obtained by replacing pseudo-vertex* S_o *by the vertex set* S_o *from which it originated). The weights* w' *of edges on* G'_c *are obtained by a transformation inverse to that performed during shrinking (thus we have* $w = w'/S_o$ *). Let* x' *be the (integer) primal solution induced by* x *on* G'_c *and* π' *the dual solution obtained on* G'_c *by assigning to the vertices in* S_o *the very dual variables which were assigned at the time when* S_o *was shrunk, and by leaving unchanged the dual variables of the vertices of* G_c *other than* S_o .

Set $G_c \leftarrow G'_c$; $w \leftarrow w'$; $x \leftarrow x'$
$\pi \leftarrow \pi'$ *and return to (c).*

(c4) The cardinality of the solution is increased by 1. End of Step p .

Set $x \leftarrow x'$; $p \leftarrow p+1$ *and return to (b).*

4.3. - Finite convergence and complexity

Each step of Algorithm 2 aiming at increasing the cardinality of the solution induced on the initial graph, there cannot be more than $\frac{N}{2}$ steps.

In order to prove finite convergence and estimate the overall complexity of the algorithm, we just have to determine an upper bound on the number of operations required at any step.

For that, we shall require the following lemma.

Lemma 1 :

Consider an arbitrary step p of Algorithm 2. Then, for all pseudo-vertices S created since the beginning of step p , it holds : $\pi_S \leq \pi_{min}(S)$

proof :

Throughout step p , only situations (B) and (C) occur, and the set of exposed vertices remains unchanged (it may happen that a pseudo-vertex be included in a pseudo-vertex, but that pseudo-vertex will remain exposed).

Each time situation (B) occurs, the pseudo-vertex Δ which is created is connected to some exposed vertex i_0 by an even (elementary) alternating chain, all the edges of which have zero reduced weight. Let $u_0 = (i_0, i_1)$, $u_1 = (i_1, i_2)$, ... $u_n = (i_n, \Delta)$ the edges successively met in a walk from i_0 to Δ .

We set : $w(L) = w_{u_0} - w_{u_1} + w_{u_2} - \dots + w_{u_{n-1}} - w_{u_n}$

and, if π is any dual solution :

$$\bar{w}_\pi(L) = \bar{w}_{u_0} - \bar{w}_{u_1} + \bar{w}_{u_2} - \dots + \bar{w}_{u_{n-1}} - \bar{w}_{u_n}$$

where, $\forall u = (i,j)$: $\bar{w}_u = w_u - \pi_i - \pi_j$

Since :

$$\bar{w}_\pi(L) = (w_{u_0} - \pi_{i_0} - \pi_{i_1}) - (w_{u_1} - \pi_{i_1} - \pi_{i_2}) + \dots - (w_{u_n} - \pi_{i_n} - \pi_\Delta)$$

we may write :

$$\bar{w}_\pi(L) = w(L) - \pi_{i_0} + \pi_\Delta$$

Now, consider some pseudo-vertex Δ , created in the course of step p and take as π , the dual variables obtained at the moment when pseudo-vertex Δ is created (thus, in particular $\pi_\Delta = \pi_{min}(\Delta)$). Since $\bar{w}_u = 0$ for all the edges in the chain L, it holds :

$$\bar{w}_\pi(L) = 0 = w(L) - \pi_{i_0} + \pi_\Delta = w(L) - \pi_{i_0} + \pi_{min}(\Delta)$$

As Algorithm 1' proceeds, the dual variables π are modified, the new values π' being such that :

$\pi'_i \leq \pi_i = \pi_{i_0}$ for all unsaturated vertices i

(property 5 § 2.4.).

In particular, for vertex i_0 : $\pi'_{i_0} \leq \pi_{i_0}$ Suppose, in a first stage, that no edge of L is included (at step p) in any pseudo-vertex created after Δ. We can then write :

$$\bar{w}_{\pi'}(L) = w(L) - \pi'_{i_0} + \pi'_\Delta$$

and, $\overline{w}_{\pi'}(L) \leq 0$ since the odd edges of chain L have reduced weight ≤ 0 (relative to π') while the even edges of L have zero reduced weight (relative to π'). Thus :

$$w(L) - \pi'_{i_0} + \pi'_{\Delta} \leq 0$$

and since $w(L) = \pi_{i_0} - \pi_{min}(\Delta)$, it follows :

$$\pi'_{\Delta} \leq \pi'_{i_0} - \pi_{i_0} + \pi_{min}(\Delta) \leq \pi_{min}(\Delta)$$

which is the desired result.

Now, suppose that some edges of L have been included in a pseudo-vertex Δ' (created after Δ at step p) and let :

$$\mu' = \ell(\pi_{min}(\Delta') - \pi'_{\Delta'})$$

the dual variable associated with S' (the vertex set corresponding to Δ'). We observe that, necessarily : $\pi'_{\Delta'} \leq \pi_{min}(\Delta')$ (this can be proved by induction, starting from the last pseudo-vertex created at step p) hence $\mu' \geq 0$.

The reduced weight of edges $u = (i,j)$ in L with both endpoints in S' is

$$\overline{w}_{u'} = w_u - \pi'_i - \pi'_j - \mu'$$

and the reduced weight of edges $u = (i,j)$ in L such that i and j do not both belong to S' is :

$$\overline{w}'_u = w_u - \pi'_i - \pi'_j .$$

In $L \cap U_{S'}$, we observe that the number of even edges exceeds the number of odd edges by 1, and therefore the reduced weight of L with respect to (π', μ') is :

$$\overline{w}_{\pi',\mu'}(L) = w(L) - \pi'_{i_0} + \pi'_{\Delta'} + \mu'$$

and this is non positive (in view of the complementary conditions). Thus we have :

$$w(L) - \pi'_{i_0} + \pi'_{\Delta} \leq w(L) - \pi'_{i_0} + \pi'_{\Delta'} + \mu' \leq 0$$

from which it follows, as above :

$$\pi'_{\Delta} \leq \pi_{min}(\Delta)$$

which is again the desired result. □

We are now in a position to prove finite convergence of Algorithm 2 and to evaluate its complexity.

Theorem 2

The complexity of Algorithm 2 is $O(N^4)$

proof :

It follows from Lemma 1 that, at each step p , no pseudo-vertex created since the beginning of step p has to be expanded. Therefore, the worst-case situation arises when, at each step p :

- all the pseudo-vertex created prior to step p need be expanded (the number of these cannot exceed $\left[\frac{N}{2}\right]$) ;
- $\left[\frac{N}{2}\right]$ new pseudo-vertices are created.

Thus, any step requires at most N longest path computations in a graph of order $2N+2$ (the lengths on the arcs all being non positive) leading to an $O(N^3)$ complexity per step, hence (since there are at most $\left[\frac{N}{2}\right]$ steps) to a total complexity $O(N^4)$. (We note here that a straightforward implementation of shrinking or expansions leads to an $O(N^2)$ complexity for each operation of this type ; since there are, at most, N shrinkings and expansions per step, and at most $\left[\frac{N}{2}\right]$ steps, the total computation cost incurred by these operations is of order $O(N^4)$ and the overall complexity of the algorithm remains $O(N^4)$). □

Though the $O(N^4)$ complexity bound given by Theorem 2 is not the best achievable (see MINOUX 1982 for several improvements in complexity of Algorithm 2) an interesting feature of the algorithm is its greater ease of implementation compared with other existing methods, and this stems from the fact that it is based on classical network flows and longest (shortest) paths procedures.

4.4. - Determining a dual optimal solution

From the discussion of section 4.3., Algorithm 2 provides, in a finite number of iterations, some shrunken graph G_c and an optimal (integer) solution x for (CMP) on G_c . The corresponding dual solution π satisfies $\pi_A \leq \pi_{min}(A)$ for every pseudo-vertex A of G_c , and the solution x^0 , induced by x on G_0 (the initial graph) is an optimum matching on G_0 .

In this section, we show how to determine an optimal dual solution (π^0, μ^0) of (CMP)' satisfying, together with x^0, the complementary conditions (C1) (C2) (C3) of § 3.2.

The current shrunken graph G_c results from the shrinking of some odd subset of vertices $S_1, S_2, \dots S_q$ corresponding to pseudo-vertices $\Delta_1, \Delta_2, \dots \Delta_q$ created (and not expanded subsequently) in the course of Algorithm 2. We may always assume (renumbering the pseudo-vertices if necessary) that

$$S_k \subset S_\ell \Rightarrow k < \ell$$

Then the solution x°, induced on G_o by the solution x on G_c, can be built up by successively expanding the pseudo-vertices in the order of decreasing indices (a pseudo-vertex thus being expanded only if all the pseudo-vertices in which it is contained have been first expanded).

We shall denote by G_ℓ the graph obtained from G_c by successively expanding $\Delta_q, \Delta_{q-1}, \dots \Delta_{k+1}$ (when $k = o$, all the pseudo-vertices have been expanded, and the resulting graph is $G_o = G$, the initial graph). For G_ℓ, we shall denote by w_u^k the weight of edge u in G_k (it is assumed that the indices of the edges remain unchanged when shrinkings or expansions occur). We shall say that a vertex j (true vertex or pseudo-vertex) is directly included in a subset S_k if $j \in S_k$ but j is not included in any $S_\ell \subset S_k$. We shall note $j \underline{\in} S_k$.

When j is directly included in S_k, we denote by $\bar{\pi}_j$ the dual variable assigned to j at the moment when Δ_k was created. Thus :

$$\pi_{min}(\Delta_k) = \min_{j \underline{\in} S_k} \{\bar{\pi}_j\}$$

Procedure 2 below then allows to simultaneously building the primal solution x° and the dual solution (π°, μ°) induced on G_o by the solutions x and π on G_c.

Procedure 2

(a) Start with $k = q$; $G_q = G_c$ *and*

$$x^q = x$$
$$\pi^q = \pi$$
$$\mu^q = 0$$

b) Current step k. *(expanding pseudo-vertex* Δ_k*). The dual variables* (π^{k-1}, μ^{k-1}) *on* G_{k-1} *are defined by :*

$\pi_i^{k-1} = \pi_i^k$ *for all vertices of* G_{k-1} *which do not belong to* S_k ;

$\pi_j^{k-1} = \bar{\pi}_j - (\tau_{min}(\Delta_k) - \pi_{\Delta_k}^k)$ *for the vertices* j *of* G_{k-1} *which belong to* S_k ;

$\mu_i^{k-1} = \mu_i^k$ *for* $i = k+1, k+2, \ldots q$

$\mu_k^{k-1} = 2(\pi_{min}(\Delta_k) - \pi_{\Delta_k}^k) - \sum_\ell \mu_\ell^k$

(summation over all indices ℓ *such that* $S_k \subset S_\ell$, $\ell \neq k$ *)*

(c) The primal solution x^{k-1} *on* G_{k-1} *is derived from* x^k *in the following way :*

Case 1 : the support of x^k *contains an edge* $u = (i, \Delta_k)$ *incident to* Δ_k *in* G_k *. This edge originates from an edge* $u = (i, j)$ *in* G_{k-1} *.* x^{k-1} *is then the matching of* G_{k-1} *deduced by appending to* x^k *the maximum matching of the odd cycle* γ_k *leaving* j *exposed.*

Case 2 : no edge in the support of x^k *is incident to* Δ_k *. In this situation,* $\pi_{\Delta_k}^k = 0$ *and there exists a vertex* $j \in S_k$ *such that* $\pi_j^{k-1} = 0$ *(this vertex is precisely such that* $\bar{\pi}_j = \pi_{min}(\Delta_k)$ *).*
Then x^{k-1} *is the matching of* G_{k-1} *obtained by appending to* x^k *the maximum matching of the odd cycle* γ_k *leaving* j *exposed.*

(d) As long as $k > 1$ *, set* $k \leftarrow k-1$ *and return to (b).*

4.5. - EDMONDS' theorem on the matching polytope

Consider the polytope $(\mathcal{P})$ **defined by the constraints of (CMP)', (cf. § 3.1.) namely :**

$$(\mathcal{P}) = \{ x \;/\; x \in \mathbb{R}^M ;\; A.x \leq \mathbb{1} ;\; F.x \leq f ;\; x \geq 0 \}$$

also referred to as the matching polytope.

For any weighting w **of the edges of** G **, Algorithm 2 of § 4.2. produces :**

- an optimal solution of (CMP)' which has all integer components ;

- a corresponding dual solution.

Thus Algorithm 2 provides a new proof of EDMONDS' Theorem which can be stated as follows :

Theorem 3 (EDMONDS 1965)

All the extreme points of the matching polytope (P) *have integer (0 - 1) components and correspond one-to-one to the characteristic vectors of matchings in* G .

4.6. - An example

We now illustrate how Algorithm 2 works on an example.

Consider the 11 - nodes, 21 - edges graph of Figure 1.

Solving the cardinality constrained continuous matching problem (CMP$[p]$) on G_o leads to integer solutions for $p = 1, 2, 3, 4$ successively - (the edges (10, 11) (7, 9) (4, 5) and (1,3) are successively selected, in that order).

The cardinality 4 matching and the corresponding dual solution thus obtained at the beginning of step 4 is given on Figure 2. (Observe that the dual variables associated with the exposed vertices are indeed all equal).

Now Step 4 begins by the determination of a longest path on the incremental graph associated with the solution of figure 2, using the double-labelling procedure of § 2.4. The longest path obtained originates at vertex 2 and corresponds to the non elementary chain (2, 4, 5, 2). We are thus in the situation where the solution of (CMP$[4+\frac{1}{2}]$) would fail to be integer and the odd subset of vertices $S_1 = \{2, 4, 5\}$ has to be shrunk. Figure 3 shows the dual variables obtained on G_o just before shrinking S_1 (observe that all edges of the odd cycle $\gamma_1 = (2, 4, 5)$ now have zero reduced weight with respect to this dual solution).

The shrunken graph $G_1 = G_o / S_1$ is shown on Figure 4 together with the corresponding weights $w^1 = w^o / S_1$ and the dual solution induced on G_1 by the dual solution of Figure 3. The pseudo-vertex corresponding to S_1 is numbered 12 and we have : $\pi_{min}(12) = 42$.

Step 4 is now resumed on G_1 . The longest path now determined by the double-labelling procedure corresponds to the edge (6,12) joining the two exposed vertices 6 and 12. A matching augmentation along this edge produces an (integer) optimal solution of cardinality 4 on G_1 (see Figure 5) which induces a cardinality 5 optimum matching on the initial graph G_0 (edges (1, 3), (2, 5), (4, 6), (7, 9), (10, 11)) End of Step 4.

We note that, in practise, the computations could be stopped at this stage. Since the initial graph has 11 vertices, a matching cannot have cardinality greater than 5, therefore the solution obtained at the end of step 4 is actually a maximum weight matching of G . However, obtaining the corresponding dual solution requires one more step to be performed (since the dual variable associated with vertex 8 is non zero).

At step 5 we start by looking for a longest path in the incremental graph associated with the solution of Figure 5. The longest path obtained corresponds to the (non elementary) chain (8, 10, 11, 8) and a new odd cycle $\gamma_2 = (8, 10, 11)$ is thereby detected. The computation should therefore be resumed on the graph G_2 deduced from G_1 by shrinking $S_2 = \{8, 10, 11\}$. Figure 6 shows the dual variables obtained on the vertices of G_1 , just before shrinking S_2 (again, observe that the edges of γ_2 have zero reduced weight with respect to this new dual solution).

The shrunken graph $G_2 = G_1 / S_2$ is shown on Figure 7 together with the edge weights $w^2 = w^1 / S_2$ and the dual solution induced by the dual solution of Figure 6. The pseudo-vertex corresponding to S_2 is numbered 13 and we have $\pi_{min}(13) = 38.5$.

Again, we look for a longest path on the incremental graph associated with the solution of Figure 7. We show on Figure 8 the dual variables which *would be obtained* after the change in dual solution. We observe that the dual variable of pseudo-vertex 12 would take the value 43.5, strictly greater than $\pi_{min}(12) = 42$.

Therefore, pseudo-vertex 12 has to be expanded. To that purpose, we first determine the convex combination of the dual solutions of Figure 7 and Figure 8 for which the dual variable for pseudo-vertex 12 is exactly 42. This solution is shown on Figure 9.

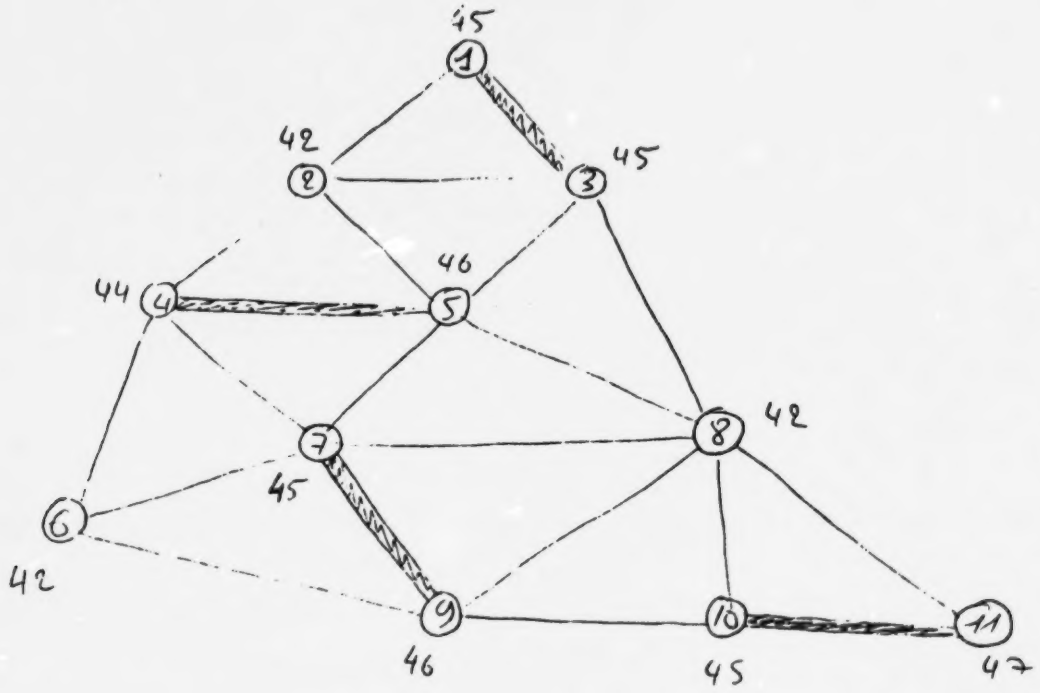


Figure 3 : The dual variables obtained on G_0 when the odd cycle $(2,4,5)$ is detected.

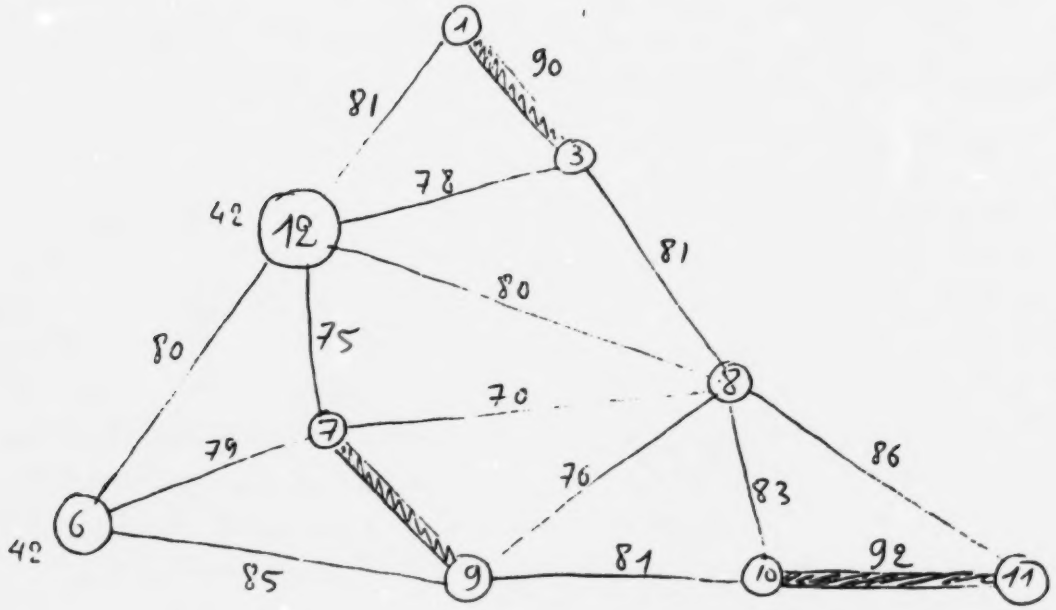


Figure 4 : The shrunken graph $G_1 = G_0/S_1$, the corresponding edge weights $w^1 = w^0/S_1$ and the dual solution induced on G_1 by the dual solution of Fig. 3.

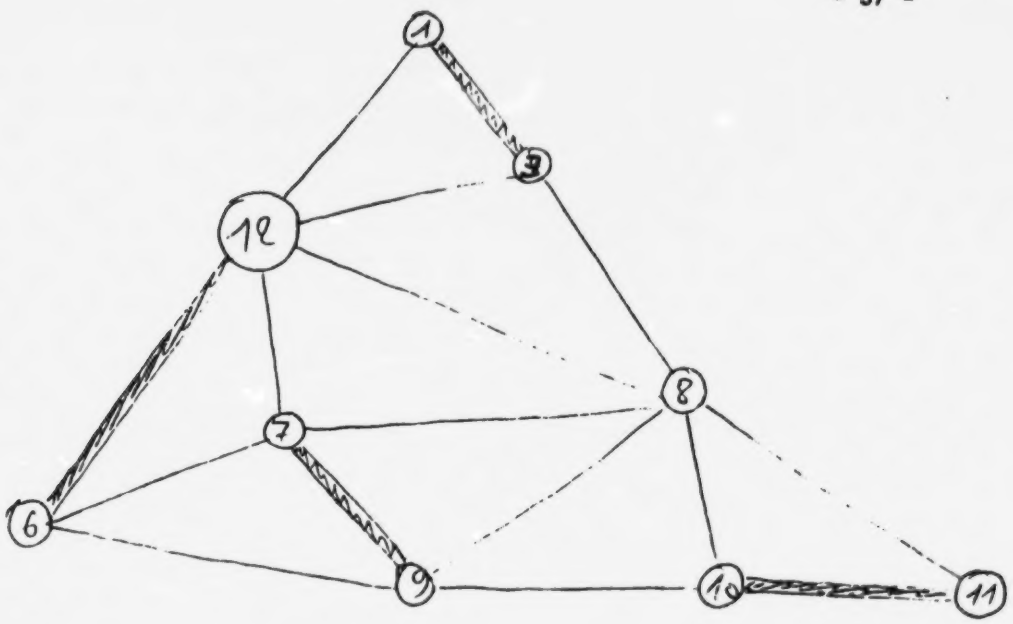


Figure 5 : The primal solution and the dual solution obtained on G_1 at the end of step 4. This optimum matching of cardinality 4 on G_1 induces an optimum matching of cardinality 5 on G_0.

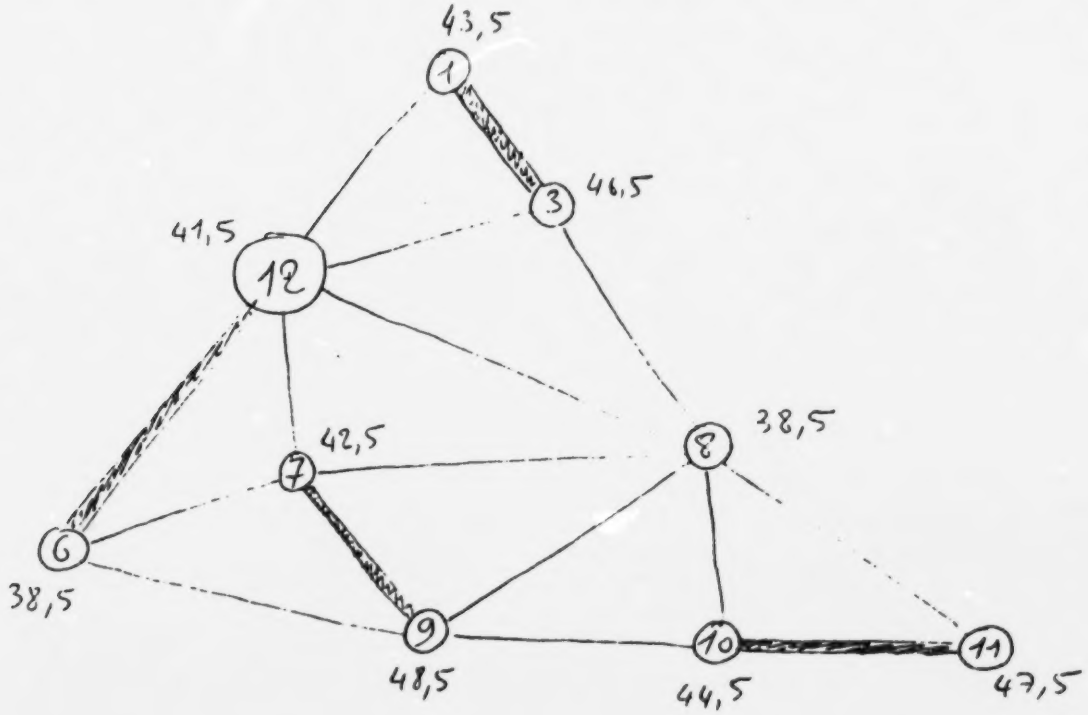


Figure 6 : The dual variables obtained on G_1 at the moment when the odd cycle $(8, 10, 11)$ is detected.

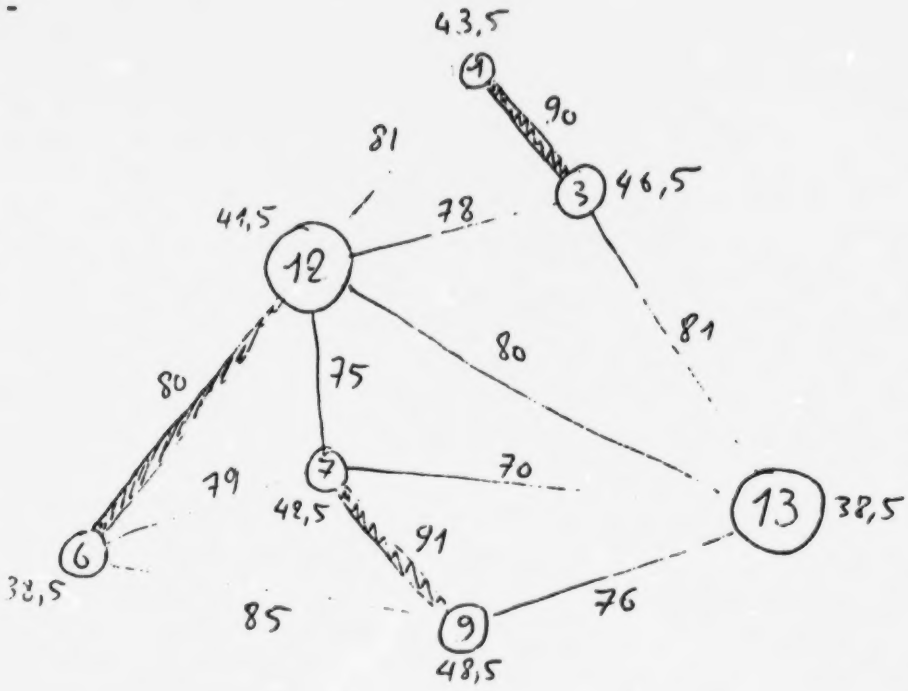


Figure 7 : The shrunken graph $G_2 = G_1 / S_2$, the weights $w^2 = w^1 / S_2$ and the dual solution induced by the dual solution of Figure 6.

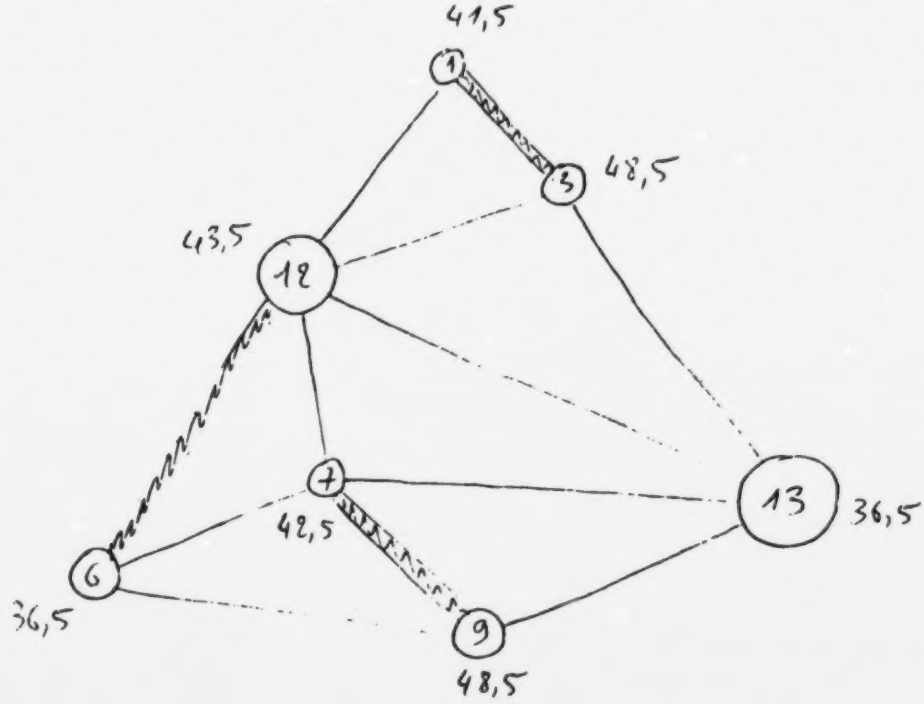
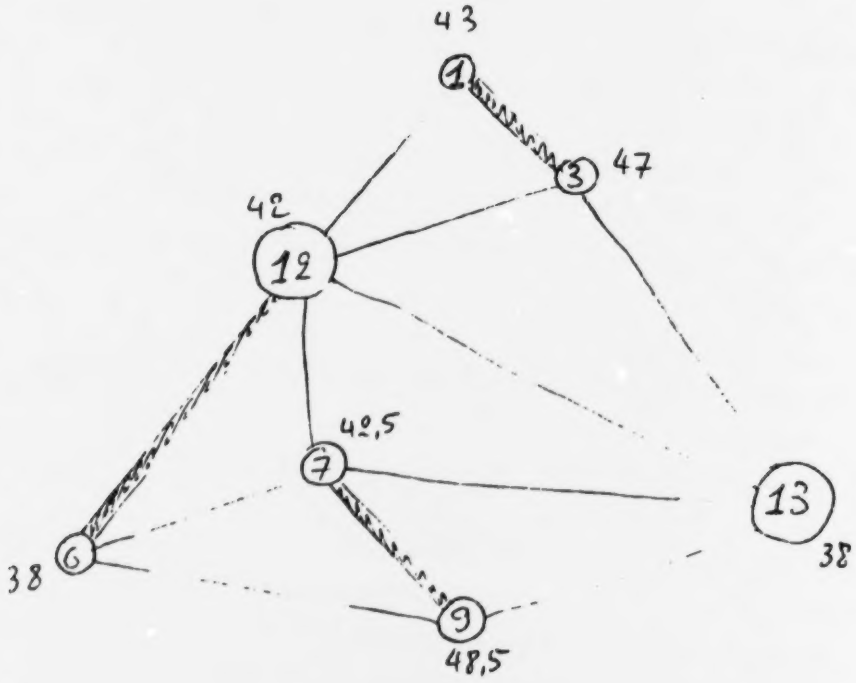


Figure 8 : The dual variables which would have been obtained after a change in dual solution on the graph of Figure 7. It is seen that the dual variable of pseudo-vertex 12 becomes strictly greater than $\pi_{min}(12) = 42$.



ure 9 : This dual solution is convex combination of the dual solutions of Figures 7 and 8, the dual variable associated with pseudo-vertex 12 being exactly $\pi_{min}(12) = 42$.

Now, pseudo-vertex 12 is expanded, and the graph of Figure 10 is obtained. The matching induced on this new graph by the primal solution of Figure 9 is obtained by choosing the maximum matching of the odd cycle $(2,4,5)$ leaving exposed vertex 4, endpoint in S_1 of edge (6, 12).

The algorithm is now resumed by looking for a longest path on the incremental graph associated with the solution of Figure 10.

The longest path obtained corresponds to the (non elementary) chain (13, 5, 2, 4, 6, 7, 9, 6, 4, 2, 5, 13) and the odd cycle $(6,7,9)$ is detected. The algorithm will thus go on on the graph G_3 deduced from the graph of Figure 10 by shrinking $S_3 = \{6,7,9\}$ (cf. Figure 11).

Again, a longest path is looked for on the graph associated with the solution of Figure 11. The (non elementary) chain (13, 3, 1, 2, 5, 13) is obtained. Figure 12 shows the dual variables obtained on G_3, and Figure 13 shows the graph G_4 deduced from G_3 by shrinking $S_4 = \{13,3,1,2,5\}$.

A longest path computation on the incremental graph associated with the solution of Figure 13 produces the (non elementary) chain (15, 14, 4, 15). The dual variables associated with the vertices 15, 14, 4 are 32, 32, 50 respectively.

The graph G_5 deduced from G_4 by shrinking the odd cycle $(15,14,4)$ has only one vertex numbered 16 (Figure 13). The primal optimal solution on G_5 is the empty matching, the dual optimal solution is $\pi_{16} = 0$.

The algorithm thus terminates at step 5 by producing an optimal solution on the current shrunken graph which is integer.

In order to get the optimum matching induced on the initial graph we just have to expand the pseudo-vertices 13, 14, 15, 16 in the order of decreasing indices.

If the edge (4, 14) is chosen as the optimum matching on the cycle (4, 14, 15) (cf. Figure 13) we find again the maximum weight matching of cardinality 5 corresponding to the matching obtained at the end of step 4 (cf. Figure 5).

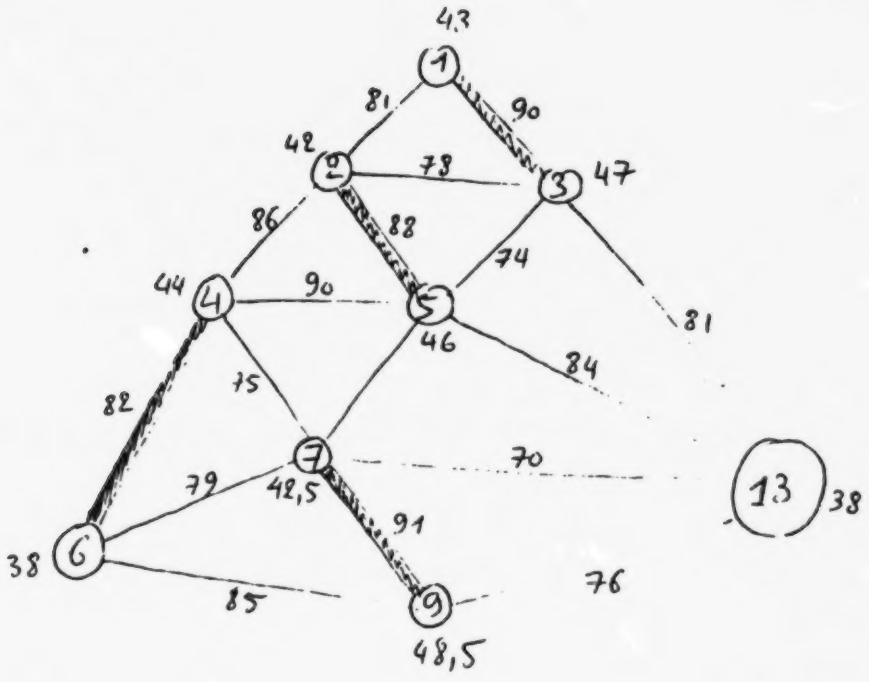


Figure 10 : The graph deduced from G_2 by expanding pseudo-vertex 12, and the primal solution induced by the current solution on G_2 . The weights have been applied the transformation inverse to that performed while shrinking. The dual solution induced by the dual solution of Figure 9 is also shown.

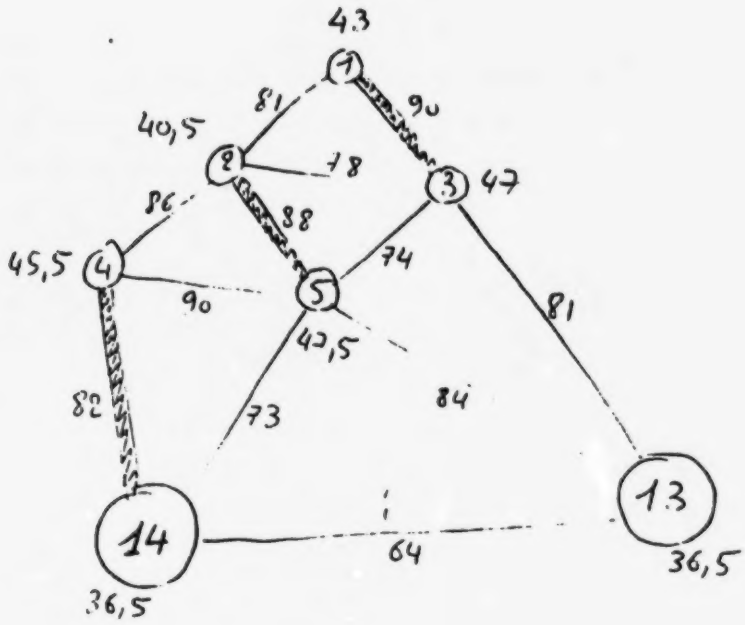


Figure 11 : The graph G_3 deduced from Figure 10 by shrinking $S_3 = \{6,7,9\}$. At the moment when shrinking occurs, the dual variables associated with 6, 7, 9 are 36.5, 42.5, 48.5 respectively.

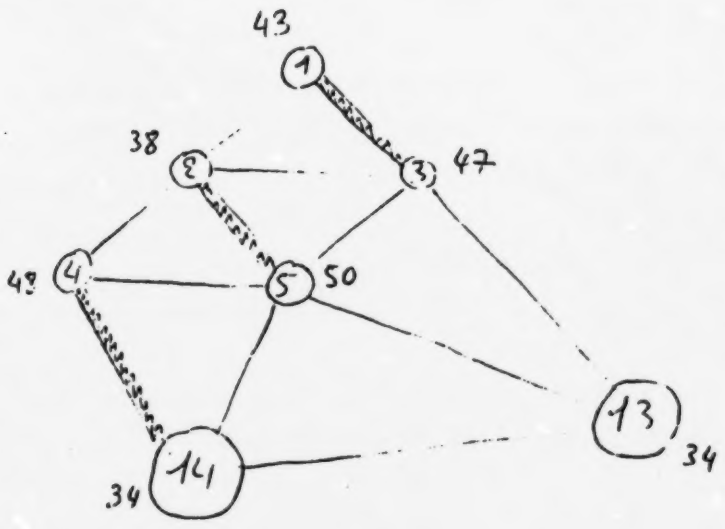


Figure 12 : The dual variables obtained on G_3 when the odd cycle $(13, 3, 1, 2, 5)$ is detected.

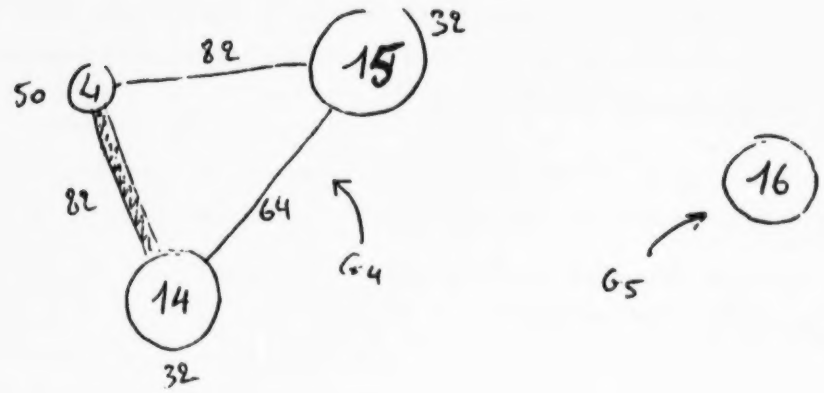


Figure 13 : The graph G_4 deduced from G_3 by shrinking $S_4 = \{13, 3, 1, 2, 5\}$ and the graph G_5 deduced from G_4 by shrinking $\{4, 14, 15\}$.

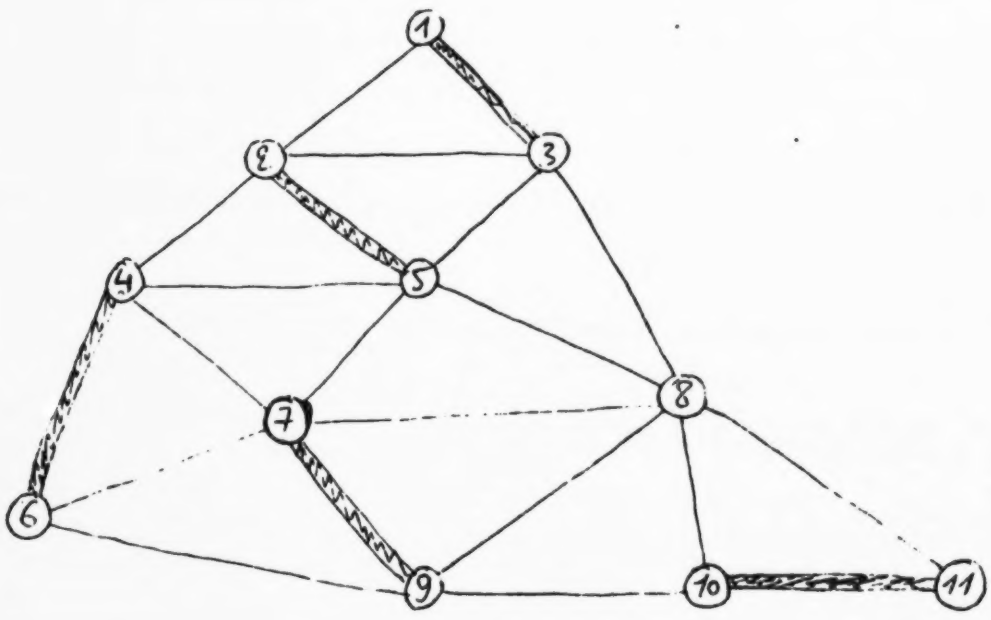


Figure 14 : The maximum weight matching induced on G_0 by the maximum weight matching (4, 14) on G_4 .

Now, the dual optimal solution (π, μ) corresponding to the maximum weight matching of Figure 14 is determined in the following way. When the algorithm terminates (here at the end of step 5) to each vertex or pseudo-vertex i is attached a number $\bar{\pi}_i$ representing :

- the current value of the dual variable associated with i if i is a vertex or a pseudo-vertex of the current shrunken graph ;
- the value of the dual variable associated with i at the moment when i was included in a pseudo-vertex.

For each pseudo-vertex s, the dual variable μ_s is obtained by :

$$\mu_s = 2(\pi_{min}(s) - \bar{\pi}_s) \quad .$$

Thus we have :

$$\mu_{16} = 2(32-0) = 64$$
$$\mu_{15} = 2(34-32) = 4$$
$$\mu_{14} = 2(36.5-32) = 9$$
$$\mu_{13} = 2(38.5-34) = 9$$

For each true vertex i, the associated dual variable π_i is then computed by

$$\pi_i = \bar{\pi}_i - \frac{1}{2} \sum \mu_s$$

where summation runs through all pseudo-vertices containing i. We thus have :

$$\pi_1 = 43-34 = 9$$
$$\pi_2 = 38-34 = 4$$
$$\pi_3 = 47-34 = 13$$
$$\pi_4 = 50-32 = 18$$
$$\pi_5 = 50-34 = 16$$
$$\pi_6 = 36.5-36.5 = 0$$
$$\pi_7 = 42.5-36.5 = 6$$
$$\pi_8 = 38.5-38.5 = 0$$
$$\pi_9 = 48.5-36.5 = 12$$
$$\pi_{10} = 44.5-38.5 = 6$$
$$\pi_{11} = 47.5-38.5 = 9$$

REFERENCES

AHO A.V., HOPCROFT J.E., ULLMAN J.D. (1974) "The Design and Analysis of Computer Algorithms"- Addison Wesley.

BALAS E. (1981) "Integer and fractional matchings"
in " *Studies on graphs and discrete programming*" (P. HANSEN ed.). North Holland p. 1-13.

BALINSKY M.L. (1969) "Labelling to obtain a maximum matching"
in :"*Combinatorial Mathematics and its Applications*" (Bose & Dowling ed.) *University North Carolina Press* p. 585.

BARTNIK G.W. (1978) "*Algorithmes de couplages dans les graphes*",
Thèse Doctorat 3ème cycle, Université Paris VI, 7 février 1978.

BERGE C. (1957) "Two theorems in graph theory"
Proc. Nat. Acad. Sc. USA, 43, p. 482.

BROWN J.R. (1974) "Shortest alternating path algorithms"
Networks 3, pp. 311-334.

BUSACKER R.G., GOWEN P.J., (1961) "A Procedure for determining a family of minimal cost network flow patterns"
O.R.O. Technical report 15, Johns Hopkins University.

CUNNINGHAM W.H., LARSH A.B. (1978) "A primal algorithm for optimum matching"
Mathematical Programming Study 8, pp 50-72.

DANTZIG G.B., BLATTNER W.O., RAO M.R. (1966) "All shortest routes from a fixed origin in graphs"
in *"Theorie des graphes"- Proceedings of the International Symposium, Rome, Juillet 1966*, Dunod Paris.

DERIGS U. (1979) "A shortest augmenting path method for solving minimal perfect matching problems"- *Report 79-06. Mathematisches Institut der Universität zu* Köln, *KOLN.*

DERIGS U. (1979) "A generalized Hungarian method for solving minimum weight perfect matching problems with algebraic objective" - *Discrete Applied Math.* 1, pp 167-180.

DIJKSTRA E.W. (1959) "A Note on two problems in connexion with graphs"
Numerische Mathematik 1, pp 269-271.

EDMONDS J.(1965a) "Paths, trees and flowers"
Canad.J. of Math. 17, pp 449-467.

EDMONDS J.(1965b) "Maximum matching and a polyhedron with 0-1 vertices"
Journal Res. Nat. Bureau Standards 69-B, n° 1-2, pp. 125-130.

EDMONDS J. (1967) "An Introduction to matching"- Lectures- Ann Arbor- University of Michigan- Michigan.

EDMONDS J., JOHNSON E.L. (1973) "Matching, Euler Tours, and the Chinese Postman"
Mathematical Programming 5, pp 88-124.

EDMONDS J., KARP R.M. (1972) "Theoretical improvements in algorithmic efficiency for network flow problems"
Journal of the A.C.M. Vol 19, n°2, pp. 248-264

EVEN S., KARIV O. (1975) "An $O(N^{2,5})$ algorithm for maximum matching in general graphs"
Proc. 16th Annual Symp. on foundations of Computer Science. I.E.E.E., New York, pp. 100-112.

FLOYD R.W. (1964) "Algorithm 245 ; treesort 3"-*Comm. ACM 7, 12, p 701.*

FORD L.R. (1956) "Network flow theory"
The Rand Corporation, p. 293.

FORD L.R., FULKERSON D.R. (1962) "*Flows in networks*"
Princeton University Press, Princeton.

GABOW H.N. (1973) "Implementation of algorithms for maximum matchings in general graphs"
Ph. D. dissertation, Comp. Sc. Dept., Stanford University, Stanford, California.

GABOW H.N. (1976) "An efficient implementation of EDMONDS' algorithm for maximum matching on graphs". *Journal ACM vol 23, n°2, pp. 221-234.*

GLOVER F. (1967) "Minimum complete matchings"
ORC- Report 67-15, University of California, Berkeley.

GOMORY R.E., HU T.C. (1961) "Multi Terminal network flows"
J. Soc. Indust. Appl. Math., 9, pp. 551-556.

GONDRAN M., MINOUX M. (1979) "*Graphes et algorithmes*"
Eyrolles, Paris.

HOLM S. (1979) "Solving the Matching Problem by cutting Planes"
Publication Department of Business and Administration, ODENSE Univ. Danemark.

HOPCROFT J.E., KARP R.M. (1973) "An $O(n^{5/2})$ algorithm for maximum matchings in bipartite graphs"
S.I.A.M. J. Computing 2, pp 225-231.

JOHNSON E. (1972) "On Shortest paths and sorting"
Proc. Annual. Conf. ACM, Boston, p 510.

LAWLER E.L. (1976) "*Combinatorial optimization : networks and matroïds*"
Holt Rinehart & Winston.

MEI-KO KWAN (1962) "Graphic Programming using odd and even points"
Chinese Math. 1, pp 273-277.

MINOUX M. (1976) "Une caractérisation des couplages de poids minimum et de cardinalité fixée dans les graphes pondérés finis"
C.R.A.S. *Paris*, t. 282, Série A, pp 353-354.

MINOUX M. (1979) *"Programmation mathématique"*
Cours de l'Ecole Nationale Supérieure de Techniques Avancées, DUNOD Paris 1982.

MINOUX M. (1979) "Couplages et b-couplages"
Chapitre 7 de *"Graphes et Algorithmes"* (M. GONDRAN, M. MINOUX) EYROLLES- Paris.

MINOUX M. (1980) "Solutions entières et solutions continues des problèmes de couplages dans les graphes pondérés finis".
C.R.A.S. Paris, t 291, Série A pp. 363-364.

MOORE E.F. (1957) "The shortest path through a maze"
Proc. Int. Symp. on theory of switching - April 2-5 1957, Harvard University Press 1959, part II, pp. 285-292.

NEMHAUSER G.L., TROTTER L.E. (1975) "Vertex packings : structural properties and algorithms"
Mathematical Programming, 8, pp 232-248.

NEMHAUSER G.L., WEBER G.M. (1979) "Optimal set partitionning, matchings and Lagrangian duality"
Naval Res. Log. Quart. 26, n°4, pp 553-563.

PADBERG M.W., RAO M.R. (1979) "Odd minimum cut sets and b-matchings"
Graduate School of Business Administration, New York University.

PICARD J.C., QUEYRANNE M. (1977) "On the Integer-valued variables in the linear vertex packing problems"
Mathematical Programming 12, pp 97-101.

ROUSSEAU J.M. (1982) "Communication orale"

TOBIN R.L. (1975) "Minimal Complete matchings and negative cycles"
Networks 5, pp 371-387.

UHRY J.P. (1975) "Sur le problème du couplage maximal"
R.A.I.R.O., Revue Verte, V-3, pp. 13-20.

WHITE L.J. (1967) "A parametric study of matchings and coverings in weighted graphs"
Doctoral Thesis, Techn. Report 06920-11-T-
Systems Engineering Lab., Dept Electrical Engineering- University of Michigan, Ann. Arbor.

WILLIAMS J.W.J. (1964) "Algorithm 232 : Heapsort"
Comm. ACM, 7,6, pp 347-348.